Universidad Pontificia Comillas

ICAI

# TFG

Study of the Feasibility, Costs, and Performance of Implementing Artificial Intelligence Algorithms Dedicated to the Segmentation of Camouflaged Living Beigns in Constrained Environments.

By,
Sanchez-Tovar Trujillo Fernando

Supervised by,
Chardon Gilles (CentraleSupélec)
Daniel Fernández Alonso (ICAI)

August, 2024

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

***Estudio de la viabilidad, impacto económico y ambiental, y desempeño de la implementación de algoritmos de inteligencia artificial dedicados a la segmentación de seres vivos camuflados en entornos restringidos.***

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2023/24 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Fernando Sánchez-Tovar Fecha: ……/ ……/ ……

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Daniel Fernández Alonso Fecha: ……/ ……/ ……

# Contents


**1 Introduction** **20**

**2 State of the Art** **21**
2.1 The Origin of modern AI and Neural Networks . . . 21
2.1.1 The Perceptron . . . 21
2.1.2 Perceptron Training . . . 22
2.1.3 Multilayer Perceptron . . . 22
2.2 Gradient Descent and Back-propagation: How do Machines Learn . . . 22
2.2.1 Gradient Descent . . . 22
2.2.2 Batch gradient descent . . . 23
2.2.3 Stochastic Gradient Descent . . . 23
2.2.4 Mini-Batch Gradient Descent . . . 24
2.2.5 Backpropagation . . . 24
2.2.6 Activation Functions . . . 26
2.2.7 Non Saturating Activation Functions . . . 27
2.3 Some NN problems . . . 28
2.3.1 Vanishing Gradients Problem . . . 28
2.3.2 Overfitting . . . 30
2.4 Convolutional Neural Networks . . . 31
2.5 MirrorNet . . . 34
2.5.1 Bio-Inspired Approach: . . . 34
2.5.2 Network Architecture: . . . 35
2.5.3 Experimental Validation: . . . 36

**3 Objectives and Problems to be Addressed** **37**
3.1 Objective 1: Understanding the Challenges in AI Model Development . . . 37
3.2 Objective 2: Analyzing the Economic, Energetic, and Environmental Impacts 37
3.3 Problems Addressed by the Project . . . 37

**4 Environmental constraints** **38**
4.0.1 Relevance of environmental impact . . . 38
4.0.2 ICT's Impact on the Environment . . . 38
4.0.3 AI's Impact on the Environment . . . 42
4.0.4 Predicting Energy Consumption & Carbon Footprint of AI . . . 44
4.0.5 Impact of LLM's . . . 46

**5 Economic Impact** **47**
5.0.1 Impact on Productivity . . . 47
5.0.2 Broader Economic Implications . . . 49
5.0.3 Impact of AI on the Labour Market . . . 50

**6 Development of the algorithms and results** **57**
6.1 Data pre-processing . . . 57
6.1.1 Camouflaged Dataset ($D_{CAM}$) . . . 58
6.1.2 Non Camouflaged Dataset ($D_{NONCAM}$) . . . 58
6.1.3 Mixed Dataset ($D_{BOTH}$) . . . 59
6.2 Classification model . . . 59
6.3 Segmentation models . . . 60
6.3.1 U-Net . . . 60

6.3.2 Attention U-Net . . . . . 62
6.3.3 R2 Attention U-Net . . . . . 64
6.4 Results and analysis . . . . . 65
6.4.1 Classification . . . . . 65
6.4.2 Segmentation Training . . . . . 66
6.4.3 Segmentation Performance . . . . . 69
6.4.4 Segmentation Illustrations . . . . . 72
6.5 Economic cost and Environmental Impact of The Models Training . . . . . 73
6.6 Improvement Avenues . . . . . 74

**Conclusion** **75**

**Bibliography** **77**

# List of Figures


1 Diagram of a Threshold Logic Unit [citation] . . . 21
2 Perceptron (one layer neural network) . . . 21
3 Gradient Descent variants . . . 24
4 Simple NN model to explain backpropagation . . . 24
5 Some activation functions and their derivatives . . . 26
6 Hierarchical extraction of features by the human eye . . . 31
7 CNN layers with rectangular local receptive fields . . . 31
8 CNN layers with rectangular local receptive fields . . . 32
9 CNN layers with rectangular local receptive fields . . . 32
10 MirrorNet framework . . . 35
11 Current status of control variables for all nine planetary boundaries (Richardson et al. [c]) . . . 40
12 Comparison of Andrae Elder, Belkir Elmeligi and Malmodin Lundén works (Environmental Impacts of Artificial Intelligence [22]) . . . 41
13 Projections of ICT's GHG emissions from 2020 studied by Freitag et al. . . . 42
14 The total amount of compute, in petaflop/s-days used to train big relatively well-known AI models over the years, OpenAI [24] . . . 43
15 Comparison of carbon emissions between BLOOM and similar LLMs. Numbers in italics have been inferred based on data provided in the papers describing the models [35]. . . . 47
16 AI adoption vs. Productivity [44] . . . 49
17 Model simulation results: demand for humans in occupation 1 (occupation being auto- mated) as a function of robot productivity, for different values of firm-level elasticity of substitution parameter . Task-level elasticity of substitution parameter t is fixed at 10 [50] . . . 53
18 Model simulation results: Exposure to robots by demographic group [50] . 55
19 Model simulation results: Exposure to software by demographic group [50] 55
20 Model simulation results: Exposure to AI by demographic group [50] . . . 56
21 An image of cat from the $D_{BOTH}$ dataset: normal image, mirrored image and rotated image, from left to right, with the corresponding masks . . . 59
22 U-Net Model (in our case k=1) . . . 61
23 Attention U-Net [56] . . . 63
24 Attention Gate [56] . . . 63
25 Different variant of convolutional and recurrent convolutional units (a) Forward convolutional units, (b) Recurrent convolutional block (c) Residual convolutional unit, and (d) Recurrent Residual convolutional units (RRCU) [57]. . . . 64
26 Confusion Matrix - Training Set . . . 65
27 Confusion Matrix - Validation Set . . . 66
28 U-Net trained on $D_{NONCAM}$ . . . 67
29 U-Net trained on $D_{BOTH}$ . . . 67
30 Attention U-Net trained on $D_{NONCAM}$ . . . 68
31 Attention U-Net trained on $D_{BOTH}$ . . . 68
32 R2 Attention U-Net trained on $D_{NONCAM}$ . . . 68
33 R2 Attention U-Net trained on $D_{BOTH}$ . . . 69
34 Prediction structure . . . 70
35 Non-representative image from the CAM dataset . . . 71

36 U-Net trained on $D_{NONCAM}$ - Segmentation of a non camouflaged image from the validation set . . . . . 72
37 U-Net trained on $D_{both}$ - Segmentation of a camouflaged image from the validation set . . . . . 72
38 R2 Attention U-Net trained on $D_{NONCAM}$ - Segmentation of a camouflaged image from the camouflages dataset (CAMO) . . . . . 73

# STUDY OF THE FEASIBILITY, COSTS, AND PERFORMANCE OF IMPLEMENTING ARTIFICIAL INTELLIGENCE ALGORITHMS DEDICATED TO THE SEGMENTATION OF CAMOUFLAGED LIVING BEIGNS IN CONSTRAINED ENVIRONMENTS

Author: Sánchez-Tovar Trujillo, Fernando

Supervisor: Fernández Alonso, Daniel

Collaborating Entity: CS Group

## SUMMARY OF THE PROJECT

Artificial Intelligence (AI) has rapidly become a transformative force in technology, aiming to replicate and exceed human cognitive abilities through advanced computational models that enable machines to interpret and respond to complex data. Among AI's critical areas, Computer Vision stands out for its focus on teaching machines to interpret visual information. This field has advanced significantly with Convolutional Neural Networks (CNNs), which excel in tasks like image classification and segmentation. Nonetheless, challenges persist, particularly in detecting camouflaged objects that blend into their surroundings, necessitating sophisticated algorithms with deep contextual understanding. This project aims to tackle these challenges by developing and analyzing CNN-based models, particularly variations of the U-Net architecture, to improve image segmentation in camouflaged environments. The project has two primary objectives: first, to assess the broader economic and environmental impacts of AI development, including the significant energy consumption and costs associated with training large-scale models , and second, to explore the technical complexities in developing AI models for complex tasks such as segmentation. By addressing these objectives, the project not only advances the technical capabilities of AI but also contributes to a deeper understanding of its environmental footprint and economic implications. This dual focus ensures that the project's outcomes are both scientifically valuable and relevant to the ongoing discourse on the sustainability and societal impact of AI.

The environmental impact of Artificial Intelligence (AI) and Information and Communication Technology (ICT) is becoming increasingly significant as these technologies expand. AI, particularly through deep learning and large language models (LLMs), is a key component of the ICT sector and demands substantial computational power, leading to high energy consumption and increased carbon emissions. This growing energy requirement also strains natural resources, such as water, which is essential for cooling data centers. ICT's environmental impact can be analyzed on three levels. Direct effects include the energy consumption and waste associated with producing, using, and disposing of ICT hardware like computers and mobile phones. While ICT has the potential to optimize processes and reduce some environmental impacts, it also contributes to negative effects such as electronic waste and increased energy demands from data centers. Additionally, ICT influences broader societal changes, which can both benefit and harm the environment—benefits like reduced transportation needs through telecommuting, and drawbacks like rebound effects that negate efficiency gains. Current assessments predict a significant increase in greenhouse gas (GHG) emissions from ICT, with expectations of a 50% rise in emissions by 2030. However, these forecasts often overlook emerging sectors such as AI, the Internet of

Things (IoT), and blockchain technologies, which are anticipated to significantly contribute to future emissions. The environmental impact of AI itself remains underexplored. AI's reliance on vast data and computational power has led to an exponential increase in energy consumption, with a 300,000-fold rise in the power used to train AI models from 2012 to 2018. Despite these growing concerns, environmental reporting in AI research is still sparse, and many studies do not comprehensively measure carbon emissions or other environmental impacts. To address these challenges, Life Cycle Assessment (LCA) methodologies are being proposed to evaluate AI's environmental impact from data acquisition to deployment. However, there is a lack of transparency and data regarding the environmental impacts of AI hardware, particularly GPUs and TPUs. There is a growing call for greater transparency and data sharing from companies to enable better environmental assessments and more sustainable AI practices. Various methods have been proposed to estimate the energy consumption and carbon footprint of AI, such as using FLOPs to gauge computational demands or employing the Power Usage Effectiveness (PUE) metric to assess data center efficiency. These approaches help quantify AI's environmental impact and guide the development of more eco-friendly AI technologies. The study of large-scale language models, such as BLOOM, further highlights the considerable environmental impact associated with training and deploying such models. BLOOM's training required over a million GPU hours and resulted in significant carbon emissions, underscoring the urgent need for more sustainable AI development practices. This study also emphasizes the ongoing energy demands of maintaining AI models post-deployment, adding to their overall environmental footprint.

Artificial Intelligence (AI) is a major force driving productivity improvements and economic growth across various sectors. By automating routine tasks, enhancing decision-making, and fostering innovation, AI significantly boosts efficiency and cost-effectiveness in organizations. Successful AI adoption hinges on strategic integration into business processes and effective management of innovation, which can lead to substantial productivity gains. However, achieving these benefits involves overcoming challenges such as workforce retraining, data privacy, and ethical considerations. The impact of AI on productivity is evident in multiple industries. In manufacturing, AI-driven predictive maintenance minimizes equipment downtimes and maintenance costs, while quality control systems ensure higher product standards. AI also enhances supply chain management by optimizing inventory and demand forecasting, which reduces overhead costs and improves operational efficiency. In the finance sector, AI revolutionizes decision-making and risk management, exemplified by institutions like Mastercard using AI for fraud detection and BlackRock employing AI for portfolio optimization. These advancements speed up transactions, reduce errors, and significantly boost productivity. Healthcare has similarly benefited from AI, with tools like IBM Watson Health improving diagnostics and treatment planning, and AI collaborations like Pfizer's with Insilico Medicine accelerating drug discovery processes. Despite the extensive productivity gains across sectors, the economic benefits of AI are not uniformly distributed, raising concerns about increasing income inequality. The disparities between those who have access to AI technologies and those who do not can hinder broader economic development. To address these challenges, it is crucial to implement strategies that promote equitable access to AI, such as education and training programs to help workers adapt to new technologies. Policymakers and industry leaders must also focus on creating environments that encourage AI innovation while addressing ethical, legal, and social challenges. To fully capitalize on AI's economic benefits, organizations should draw on best practices from various industries, including effective AI integration, workforce retraining, data security measures, and regulatory

compliance. By addressing these aspects, organizations can ensure that AI technologies contribute to sustained economic growth and improved competitiveness in an increasingly technological world.

AI's impact on the labor market is profound, influencing job displacement and income inequality. This project, using information from Michael Webb's research, explores how AI, particularly machine learning algorithms, is reshaping the labor market by automating tasks across various occupations. Webb's methodology assesses the "exposure" of different jobs to AI-driven automation, providing insights into which roles are most vulnerable and estimating AI's impact on wage inequality. Webb's findings indicate that high-skilled occupations are particularly exposed to AI, suggesting that AI's effects differ significantly from previous technologies like software and robots. High-wage, education-intensive jobs may experience greater displacement as AI automates tasks traditionally performed by highly skilled professionals. This could lead to increased wage inequality at the top of the income distribution, even as it reduces inequality in mid-level occupations. The study also highlights how AI could either reduce or increase overall labor demand depending on the context, as AI-driven automation and broader market effects of increased productivity lead to complex outcomes. Historically, technologies like robots and software primarily disrupted lower-wage manual jobs and middle-wage technical roles, with minimal impact on high-skill, well-paid jobs. However, AI is expected to impact high-wage jobs more significantly, potentially leading to greater disparities in income levels. This underscores the need for policymakers and businesses to anticipate and manage the labor market changes driven by AI, focusing on strategies that mitigate the risks of increased wage inequality while fostering job creation and economic stability.

The development of AI models capable of segmenting camouflaged animals in constrained environments illustrates the complexity involved in creating advanced AI systems. This project, focused on developing a Convolutional Neural Network (CNN) algorithm for this purpose, reveals the numerous steps required, including data preprocessing, model selection, fine-tuning, and extensive training. Each of these steps underscores the challenges and sophistication required to build AI models that can effectively perform complex tasks like image segmentation.

The initial step in developing the AI model involved meticulous data preprocessing, which is essential for training CNNs. The data consisted of images formatted into multi-dimensional numpy arrays, where each image was paired with a corresponding mask that delineated the desired output (i.e., the animal's silhouette). The images were resized to 256x256x3 arrays, representing RGB channels, while the masks were resized to 256x256x1 arrays, representing binary images where the animal was highlighted against a black background. This preprocessing step was crucial to ensure that the images and masks could be accurately fed into the neural networks during training. Two main datasets were employed:

- Camouflaged Dataset (DCAMCAM): Provided by CS Group, this dataset included various types of images, but only the camouflaged ones (3,758 images) were used after filtering out corrupted and non-camouflaged images.
- Non-Camouflaged Dataset (DNONCAMNONCAM): The Oxford-IIIT Pet dataset was used as a simpler starting point to develop the model, consisting of 7,383 images of dogs and cats with corresponding masks. This dataset allowed the development of robust algorithms before tackling the more challenging task of segmenting camouflaged images.

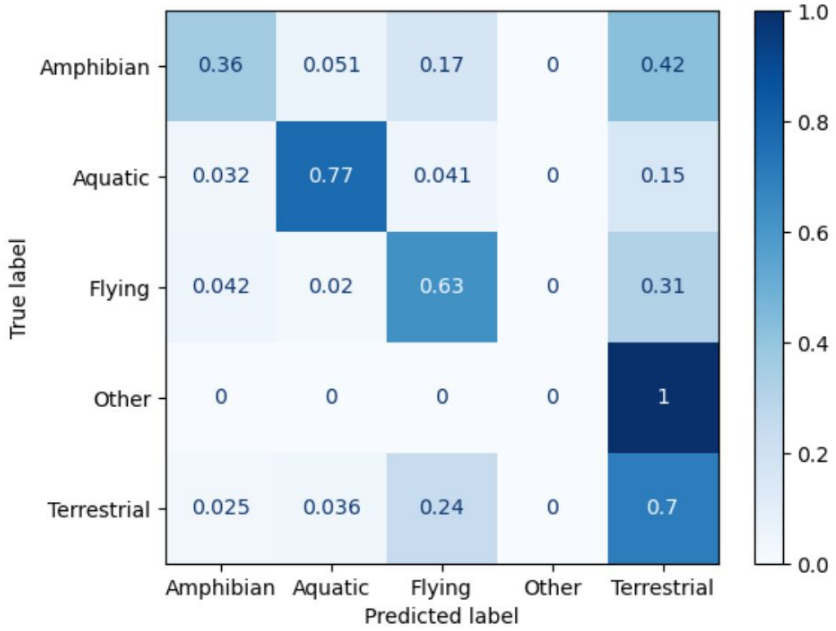
Additionally, a Mixed Dataset (DBOTHBOTH) was created by combining and augmenting images from both datasets, incorporating a total of 1,201 images to improve model robustness.

The project utilized three key CNN architectures: ResNet50, U-Net, and its variations (Attention U-Net and R2 Attention U-Net), each selected for specific reasons and providing different levels of performance and complexity.

- ResNet50: ResNet50 is a deep CNN with approximately 23.6 million parameters, known for its ability to address the vanishing gradient problem through the introduction of residual blocks and skip connections. These features allow gradients to flow more effectively during training, making it suitable for deep networks. In this project, ResNet50 was fine-tuned to classify camouflaged images into five classes: Amphibian, Aquatic, Flying, Terrestrial, and Other. Only the final layers of the pre-trained model were trained on the camouflaged dataset, preserving its pre-existing feature detection capabilities.
- U-Net: U-Net is a CNN architecture originally designed for biomedical image segmentation. It features a U-shaped structure with a contracting path that captures contextual information and an expansive path that enables precise localization. This model was chosen due to its ability to perform accurate segmentation with relatively small datasets, making it ideal for the camouflaged animal images. It has 7.8 million parameters and was trained to output a binary image where the animal was segmented from the background.
- Attention U-Net: A variant of U-Net, Attention U-Net incorporates attention gates to focus on relevant regions of the image, improving the model's generalization by reducing irrelevant activations. The attention gates help the model to concentrate on areas of the image where the camouflaged animal is likely to be, thereby enhancing segmentation performance. The integrated model has 9.3 million parameters.
- R2 Attention U-Net: This model combines the advantages of Attention U-Net with those of R2U-Net, which integrates Residual Recurrent Convolution blocks. These blocks allow the model to handle spatial and temporal dependencies more effectively. The R2 Attention U-Net model was specifically implemented to further improve the segmentation of challenging camouflaged images by leveraging both the attention mechanisms and the residual recurrent features.

The complexity of developing these models lies not only in the architecture and training but also in the data processing and model selection stages. The careful curation of datasets, the need for augmentation to overcome data scarcity, and the selection of appropriate models all contribute to the overall challenge. Each model requires extensive computational resources, and careful monitoring to avoid issues like overfitting or vanishing gradients.

The neural network models were trained on a high-performance computer provided by CentraleSupélec, equipped with four NVIDIA A100 GPUs, an AMD 7742 CPU, and other advanced components, with a total cost ranging from 55,000 € to 80,000 €. Operating at 30% capacity due to shared usage, the system consumed approximately 8.25 kWh over nearly 50 hours, resulting in a carbon footprint of 462.2 grams of CO and an electricity cost of under 0.4 €. These minimal costs and environmental impacts are significantly lower compared to the resources required for training large-scale models like GPT-4, which demand far greater energy, emissions, and financial investment




Confusion Matrix - Validation Set

Regarding the results of the classification algorithm, due to the dataset's complexity given the camouflaged animals and the variety of species, the classification model was trained for 10 epochs but halted at the 6th epoch due to early stopping, with the model failing to improve beyond this point. The confusion matrix from the training data showed that images from "Amphibian" and "Flying" classes were frequently misclassified as "Terrestrial," reflecting environmental similarities. Additionally, 81% of the "Other" class images were misclassified as "Terrestrial," likely due to incorrect categorization by the dataset creator. Validation set results were more pronounced in misclassification: only 36% of "Amphibian" images were correctly classified, with 42% mislabeled as "Terrestrial," and all images in the "Other" category were misclassified. The model achieved a validation accuracy of 66%, which could potentially be improved by removing or better redistributing the "Other" class images.

| **Model** | **Number of parameters** | **Dataset** | **Training Time (h:min:s)** | **Number of epochs** |
|---|---|---|---|---|
| UNET | 7,771,873 | D_NONCAM | 06:57:34 | 12 |
| | | D_BOTH | 01:04:44 | 12 |
| Attention | 9,344,841 | D_NONCAM | 17:15:33 | 20 |
| | | D_BOTH | 02:41:39 | 20 |
| Residual | 9,787,497 | D_NONCAM | 18:50:35 | 20 |
| | | D_BOTH | 03:04:55 | 20 |

Segmentation models (U-Net, Attention U-Net, and R2 Attention U-Net) were trained on two datasets: $D_{NONCAM}$ (non-camouflaged images) and $D_{BOTH}$ (mixed images). Training times varied significantly due to the size of the datasets and the complexity of the models. For instance, Attention U-Net and R2 Attention U-Net, with additional parameters and complexity, took longer to train compared to the basic U-Net. U-Net training was cut short by early stopping due to a lack of improvement in validation loss. Training and validation loss curves for U-Net showed effective learning but indicated overfitting towards the end. Attention U-Net and R2 Attention U-Net demonstrated consistent improvement in validation loss, with Attention U-Net experiencing a temporary increase in loss likely due to the stochastic nature of the Adam optimizer. Both models

showed promise for better results with extended training.

| Model | Number of parameters | Training Dataset | Evaluation Dataset | IoU |
|---|---|---|---|---|
| **UNET** | 7,771,873 | D_NONCAM | CAM | 0.452 |
| | | | Training Set | 0.866 |
| | | | Validation Set | 0.832 |
| | | D_BOTH | Training Set | 0.646 |
| | | | Validation Set | 0.637 |
| **Attention** | 9,344,841 | D_NONCAM | CAM | 0.508 |
| | | | Training Set | 0.865 |
| | | | Validation Set | 0.831 |
| | | D_BOTH | Training Set | 0.592 |
| | | | Validation Set | 0.590 |
| **Residual** | 9,787,497 | D_NONCAM | CAM | 0.520 |
| | | | Training Set | 0.808 |
| | | | Validation Set | 0.757 |
| | | D_BOTH | Training Set | 0.559 |
| | | | Validation Set | 0.532 |

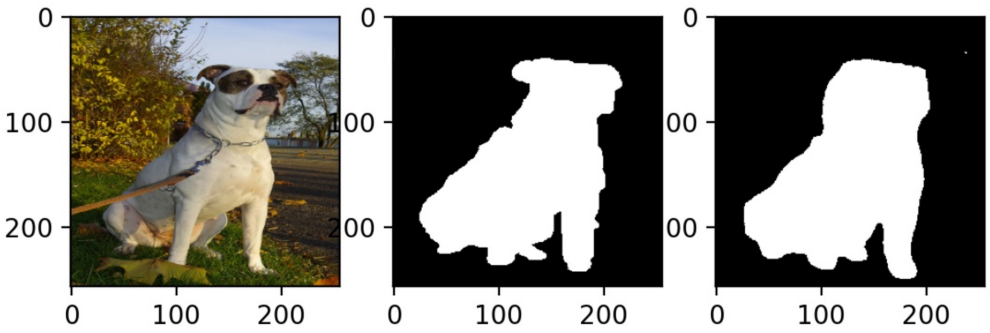
The segmentation performance was evaluated using Intersection over Union (IoU) as a metric. Models trained on $D_{NONCAM}$ and evaluated on the CAM dataset of camouflaged animals showed:

- U-Net: IoU = 0.452
- Attention U-Net: IoU = 0.508
- R2 Attention U-Net: IoU = 0.520

Models trained on $D_{BOTH}$ and evaluated on the validation set showed:

- U-Net: IoU = 0.637
- Attention U-Net: IoU = 0.590
- R2 Attention U-Net: IoU = 0.532

Among the models trained on $D_{NONCAM}$, R2 Attention U-Net had the highest IoU for camouflaged animals, followed by Attention U-Net and then U-Net. Conversely, for models trained on $D_{BOTH}$, U-Net performed the best, followed by Attention U-Net and R2 Attention U-Net. These results highlight the models' effectiveness on non-camouflaged images but reveal challenges in segmenting camouflaged animals. Potential issues include the complexity of camouflaged images, insufficient camouflaged data, and limitations of the model architecture in handling such cases. Despite these challenges, models like R2 Attention U-Net showed promising generalization capabilities and the potential for further improvement with additional training and adjustments to the model structure.




U-Net trained on $D_{NONCAM}$ - Segmentation of a non camouflaged image from the validation set (Original Image / Mask / Prediction)

Visual examples of segmentation results demonstrate the models' performance. For non-camouflaged images, all models performed well, accurately segmenting the animals. In contrast, segmentation of camouflaged images revealed limitations, with models like U-Net struggling to distinguish the animal from the background and R2 Attention U-Net showing potential for improvement in prediction shape and detail recognition. These observations underscore the need for enhanced model capabilities and additional camouflaged data to improve segmentation accuracy in complex scenarios.

To conclude, Artificial intelligence (AI) is rapidly evolving as a transformative technology with significant implications for the future. This project, extending a study conducted in collaboration with the CS Group, examines the economic and environmental impacts of AI while also tackling the specific challenge of camouflaged animal segmentation to explore the complexities of developing sophisticated AI models. The investigation into various Convolutional Neural Network (CNN) architectures, such as U-Net, Attention U-Net, and R2 Attention U-Net, reveals that while these models excel at segmenting non-camouflaged animals, they struggle with camouflaged images, highlighting the difficulty of designing AI systems for nuanced real-world scenarios. The R2 Attention U-Net model showed promise and indicates that enhanced architectures and computational resources could improve performance. Economically, AI is expected to boost productivity and drive growth, yet it may also impact labor markets by shifting demand toward highly educated, well-paid jobs, affecting the income inequality distribution . Furthermore, the environmental impact of AI's growing computational needs raises concerns about energy consumption and carbon emissions. This project emphasizes the need for better methodologies to assess and mitigate the environmental costs of AI and highlights the importance of balancing the technology's benefits with its economic and environmental challenges. As AI continues to advance, thoughtful consideration of its impacts and ongoing refinement of development practices will be essential for harnessing its potential while addressing its associated challenges.

# ESTUDIO DE LA VIABILIDAD, COSTES Y RENDIMIENTO DE LA IMPLEMENTACIÓN DE ALGORITMOS DE INTELIGENCIA ARTIFICIAL DEDICADOS A LA SEGMENTACIÓN DE SERES VIVOS CAMUFLADOS EN ENTORNOS RESTRINGIDOS

Autor: Sánchez-Tovar Trujillo, Fernando

Supervisor: Fernández Alonso, Daniel

Entidad Colaboradora: CS Group

## RESUMEN DEL PROYECTO

La Inteligencia Artificial (IA) se ha convertido rápidamente en una fuerza transformadora en la tecnología, que pretende imitar e incluso superar las capacidades lógicas y de raciocinio humanas mediante modelso computacionales complejos capaces de analizar e interpretar datos complejos. Entere los sectores de IA que más se han desarrollado en los últimos años destaca la Visión Artificial o Visión por Computadora, que pretende enseñar a las máquinas a interpretar información visual. Este campo ha avanzado significativamente con las Redes Neuronales Convolucionales (CNNs), que destacan en tareas como la clasificación y segmentación de imágenes. Sin embargo, persisten desafíos, particularmente en la detección de objetos camuflados que se mezclan con su entorno, lo que requiere algoritmos sofisticados con una profunda comprensión contextual. Este proyecto tiene como objetivo abordar estos desafíos desarrollando y analizando modelos basados en CNN, particularmente variaciones de la arquitectura U-Net, para mejorar la segmentación de imágenes en entornos camuflados. El proyecto tiene dos objetivos principales: primero, evaluar los impactos económicos y ambientales del desarrollo de IA, incluyendo el significativo consumo de energía y los costes asociados con el entrenamiento de modelos a gran escala, y segundo, explorar las complejidades técnicas en el desarrollo de modelos de IA para tareas complejas como la segmentación. Al abordar estos objetivos, el proyecto no solo avanza en las capacidades técnicas de la IA, sino que también contribuye a una comprensión más profunda de su huella ambiental e implicaciones económicas. Este enfoque dual asegura que los resultados del proyecto sean tanto científicamente valiosos como relevantes para el discurso sobre la sostenibilidad y el impacto social de la IA.

El impacto ambiental de la Inteligencia Artificial (IA) y las Tecnologías de la Información y Comunicación (TIC) está adquiriendo una relevancia creciente a medida que estas tecnologías se expanden. La IA, particularmente a través del 'deep learning' o aprendizaje profundo y los 'Large Language Models' (LLMs) o modelos grandes de lenguaje, es un componente clave del sector TIC y demanda una considerable capacidad de computación. Esto lleva a un alto consumo de energía y un aumento en las emisiones de carbono. Este creciente requisito de energía también ejerce presión sobre los recursos naturales, como el agua, que es esencial para la refrigeración de los centros de datos. El impacto ambiental de las TIC se puede analizar en tres niveles. Los efectos directos incluyen el consumo de energía y los desechos asociados con la producción, el uso y la eliminación de hardware TIC como computadoras y teléfonos móviles. Aunque las TIC tienen el potencial de optimizar procesos y reducir algunos impactos ambientales, también contribuyen mediante efectos negativos como los desechos electrónicos y el aumento de la demanda energética de los centros de datos. Además, las TIC influyen en cambios sociales más amplios, que pueden

beneficiar o perjudicar al medio ambiente. Beneficios como la reducción de la necesidad de transporte a través del teletrabajo, y desventajas como los efectos rebote que anulan las ganancias en eficiencia. Los estudios actuales predicen un aumento significativo en las emisiones de gases de efecto invernadero (GEI) provenientes de las TIC, con expectativas de un incremento del 50% en las emisiones para 2030. Sin embargo, estas previsiones a menudo pasan por alto sectores emergentes como la IA, el Internet de las Cosas (IoT) y las tecnologías blockchain, que se espera que contribuyan significativamente a las futuras emisiones. El impacto ambiental de la IA sigue sin explorarse en profundidad. La dependencia de la IA en enormes cantidades de datos y en la capacidad de computación ha llevado a un aumento exponencial en el consumo de energía, con un aumento de 300,000 veces en la capacidad utilizada para entrenar modelos de IA desde 2012 hasta 2018. A pesar de estas crecientes preocupaciones, los informes ambientales en la investigación de IA aún son escasos, y muchos estudios no miden de manera exhaustiva las emisiones de carbono u otros impactos ambientales. Para abordar estos desafíos, se están proponiendo metodologías basadas en la Evaluación del Ciclo de Vida para evaluar el impacto ambiental de la IA desde la adquisición de datos hasta su uso por los consumidores últimos de la misma. Sin embargo, hay una falta de transparencia y datos respecto a los impactos ambientales del hardware de IA, particularmente GPUs y TPUs. Existe una creciente demanda de mayor transparencia por parte de las empresas para permitir mejores evaluaciones ambientales y prácticas de IA más sostenibles. Se han propuesto varios métodos para estimar el consumo de energía y la huella de carbono de la IA, como el uso de FLOPs para medir las demandas computacionales o el empleo de la métrica 'Power Usage Effectiveness' (PUE) para evaluar la eficiencia de los centros de datos. Estos enfoques ayudan a cuantificar el impacto ambiental y guiar el desarrollo de tecnologías de IA más ecológicas. El estudio de modelos de lenguaje a gran escala, como BLOOM, resalta aún más el considerable impacto ambiental asociado con el entrenamiento y la implementación de dichos modelos. El entrenamiento de BLOOM requirió más de un millón de horas de GPU y resultó en emisiones significativas de carbono, subrayando la necesidad urgente de prácticas de desarrollo de IA más sostenibles.

La Inteligencia Artificial (IA) es una tecnología capaz de impulsar la mejora de la productividad y el crecimiento económico en diversos sectores. Al automatizar tareas rutinarias, mejorar la toma de decisiones y fomentar la innovación, la IA aumenta significativamente la eficiencia y la rentabilidad en las organizaciones. La adopción exitosa de la IA depende de su integración estratégica en los procesos de negocio y la gestión efectiva de la innovación, lo que puede llevar a importantes ganancias en productividad. Sin embargo, lograr estos beneficios implica superar desafíos como la formación de los trabajadores, la privacidad de los datos y consideraciones éticas. El impacto de la IA en la productividad es evidente en múltiples industrias. En la industria manufacturera, el mantenimiento predictivo impulsado por IA minimiza los tiempos de inactividad del equipo y los costes de mantenimiento, mientras que los sistemas de control de calidad aseguran estándares de producto más altos. La IA también mejora la gestión de la cadena de suministro al optimizar el inventario y la previsión de la demanda, lo que reduce los costes generales y mejora la eficiencia operativa. En el sector financiero, la IA revoluciona la toma de decisiones y la gestión de riesgos, ejemplificado por instituciones como Mastercard utilizando IA para la detección de fraude y BlackRock empleando IA para la optimización de carteras. Estos avances aceleran las transacciones, reducen errores y aumentan significativamente la productividad. La atención médica también ha beneficiado de la IA, con herramientas como IBM Watson Health mejorando los diagnósticos y la planificación del tratamiento, y colaboraciones de IA como la de Pfizer con Insilico Medicine acelerando los procesos de descubrimiento

de fármacos. A pesar de las extensas ganancias en productividad a través de diversos sectores, los beneficios económicos de la IA no se distribuyen uniformemente, lo que genera preocupaciones sobre el aumento de la desigualdad económica. Las disparidades entre quienes tienen acceso a las tecnologías de IA y quienes no lo tienen podrían obstaculizar el desarrollo económico. Para abordar estos desafíos, es crucial implementar estrategias que promuevan un acceso equitativo a la IA, como programas de educación y capacitación para ayudar a los trabajadores a adaptarse a las nuevas tecnologías. Los políticos y los líderes de la industria también deben centrarse en crear entornos que fomenten la innovación en IA mientras abordan desafíos éticos, legales y sociales. Para capitalizar completamente los beneficios económicos de la IA, las organizaciones deben basarse en las mejores prácticas de diversas industrias, incluyendo la integración efectiva, la formación de la fuerza laboral, medidas de seguridad de datos y cumplimiento normativo. Al abordar estos aspectos, las organizaciones pueden asegurar que las tecnologías de IA contribuyan a un crecimiento económico sostenible y a una mayor competitividad en un mundo cada vez más tecnológico.

El impacto de la IA en el mercado laboral es profundo, influyendo en el desplazamiento de empleos y las desigualdades económicas. Este proyecto, usando información de la investigación llevada a cabo por Michael Webb, explora cómo la IA, particularmente los algoritmos de aprendizaje automático, está remodelando el mercado laboral al automatizar tareas en diversas ocupaciones. La metodología de Webb evalúa la "exposición" de diferentes trabajos a la automatización impulsada por la IA, proporcionando información sobre qué roles son más vulnerables y estimando el impacto de la IA en la desigualdad salarial. Los hallazgos de Webb indican que las ocupaciones altamente cualificadas están particularmente expuestas a la IA, sugiriendo que los efectos de la IA difieren significativamente de tecnologías anteriores como el software y los robots. Los trabajos de alta remuneración y que requieren mucha educación pueden experimentar un mayor desplazamiento a medida que la IA automatiza tareas tradicionalmente realizadas por profesionales altamente cualificados. Esto podría llevar a una mayor desigualdad salarial en la parte superior de la distribución de ingresos, incluso cuando reduce la desigualdad en las ocupaciones de nivel medio. El estudio también destaca cómo la IA podría reducir o aumentar la demanda laboral general dependiendo del contexto, ya que la automatización impulsada por la IA y los efectos causados por el aumento de la productividad conducen a resultados complejos. Históricamente, tecnologías como los robots y el software han perturbado principalmente trabajos manuales de baja remuneración y roles técnicos de remuneración media, con un impacto mínimo en trabajos de alta complejidad y bien remunerados. Sin embargo, se espera que la IA impacte de manera más significativa los trabajos de alta remuneración, lo que podría llevar a mayores disparidades en los niveles de ingresos. Esto subraya la necesidad de que los políticos y las empresas anticipen y gestionen los cambios en el mercado laboral impulsados por la IA, enfocándose en estrategias que mitiguen los riesgos de aumento de la desigualdad salarial mientras fomentan la creación de empleo y la estabilidad económica.

El desarrollo de modelos de IA capaces de segmentar animales camuflados en entornos restringidos ilustra la complejidad involucrada en la creación de sistemas avanzados de IA. Este proyecto, centrado en el desarrollo de un algoritmo basado en las Redes Neuronales Convolucionales (CNN), ejemplifica y explora cada una de las etapas necesarias para desarrollar un modelo de IA complejo: el preprocesamiento de datos, la selección y creación de modelos, y el entrenamiento extensivo. Cada uno de estos pasos subraya los desafíos y la sofisticación necesarios para construir modelos de IA que puedan realizar

eficazmente tareas complejas como la segmentación de imágenes.

El primer paso en el desarrollo del modelo de IA involucró un meticuloso preprocesamiento de datos, esencial para el entrenamiento de CNNs. Los datos consistieron en imágenes, que hubo que formatear en arrays multidimensionales de la biblioteca numpy, donde cada imagen se emparejó con una máscara correspondiente que delimitaba la salida deseada (es decir, la silueta del animal). Las imágenes fueron redimensionadas en arrays de 256x256x3, que representan los canales RGB, mientras que las máscaras se redimensionaron a arrays de 256x256x1, que representan imágenes binarias donde el animal estaba resaltado contra un fondo negro. Este paso de preprocesamiento fue crucial para asegurar que las imágenes y las máscaras pudieran ser introducidas con precisión en las redes neuronales durante el entrenamiento.

Se emplearon dos conjuntos de datos principales:

- Conjunto de Datos Camuflados ($D_{CAM}$): Proporcionado por CS Group, este conjunto incluía varios tipos de imágenes, pero solo se utilizaron las camufladas (3,758 imágenes) después de filtrar las imágenes corruptas y no camufladas.
- Conjunto de Datos No Camuflados ($D_{NONCAM}$): Se utilizó el conjunto de datos Oxford-IIIT Pet como un punto de partida más simple para desarrollar el modelo, compuesto por 7,383 imágenes de perros y gatos con sus máscaras correspondientes. Este conjunto permitió el desarrollo de algoritmos robustos antes de abordar la tarea más compleja de segmentar animales camuflados.

Además, se creó un Conjunto de Datos Mixto ($D_{BOTH}$) combinando imágenes de ambos conjuntos y creando de forma artificial nuevas imágenes camufladas basadas en las de $D_{CAM}$, incorporando un total de 1,201 imágenes para mejorar la robustez del modelo.

El proyecto utilizó tres arquitecturas de CNN: ResNet50, U-Net y sus variaciones (Attention U-Net y R2 Attention U-Net), cada una seleccionada por razones específicas y proporcionando distintos niveles de rendimiento y complejidad.

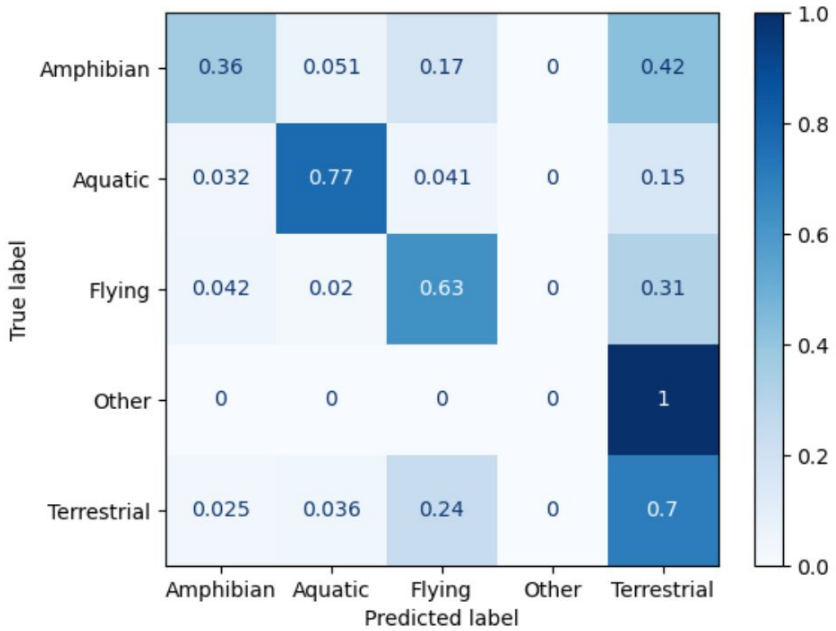
- ResNet50: ResNet50 es una CNN profunda con aproximadamente 23.6 millones de parámetros, conocida por su capacidad para abordar el 'Vanishing Gradients Problem' mediante la introducción de bloques residuales y conexiones de salto. Estas características permiten que los gradientes fluyan de manera más efectiva durante el entrenamiento, lo que la hace adecuada para redes profundas. En este proyecto, ResNet50 se ajustó para clasificar imágenes camufladas en cinco clases: Anfibio, Acuático, Volador, Terrestre y Otro. Solo las capas finales del modelo preentrenado se entrenaron en el conjunto de datos camuflado, preservando sus capacidades de detección de características preexistentes.
- U-Net: U-Net es una arquitectura de CNN diseñada originalmente para la segmentación de imágenes biomédicas. Presenta una estructura en forma de U con un camino de contracción que captura información contextual y un camino expansivo que permite una localización precisa. Este modelo se eligió debido a su capacidad para realizar segmentaciones precisas con conjuntos de datos relativamente pequeños, lo que lo hace ideal para las imágenes de animales camuflados. Tiene 7.8 millones de parámetros y se entrenó para producir una imagen binaria donde el animal estaba segmentado (o resaltado con respecto al fondo).
- Attention U-Net: Una variante de U-Net, Attention U-Net incorpora puertas de atención para centrarse en regiones relevantes de la imagen, mejorando la generalización del modelo al reducir las activaciones en zonas irrelevantes. Las puertas de atención

ayudan al modelo a concentrarse en las zonas de la imagen donde es más probable que esté el animal camuflado, mejorando así el rendimiento de la segmentación. El modelo integrado tiene 9.3 millones de parámetros.

- R2 Attention U-Net: Este modelo combina las ventajas de Attention U-Net con las de R2U-Net, que integra bloques de Convolución Recurrente Residual. Estos bloques permiten que el modelo maneje las dependencias espaciales y temporales de manera más efectiva. El modelo R2 Attention U-Net se implementó específicamente para mejorar aún más la segmentación de imágenes camufladas aprovechando tanto los mecanismos de atención como las características residuales recurrentes.

La complejidad de desarrollar estos modelos radica no solo en la arquitectura y el entrenamiento, sino también en las etapas de procesamiento de datos y la selección de modelos. Cada modelo requiere recursos computacionales extensivos y una supervisión cuidadosa para evitar problemas como el 'overfitting' o el 'Vanishing Gradients Problem'.

Los modelos se entrenaron en un ordenador de alto rendimiento proporcionado por CentraleSupélec, equipado con cuatro GPUs NVIDIA A100, un CPU AMD 7742 y otros componentes avanzados, con un coste total que varía entre 55,000 € y 80,000 €. Funcionando al 30% de su capacidad debido a un uso compartido, el sistema consumió aproximadamente 8.25 kWh durante casi 50 horas, resultando en una huella de carbono de 462.2 gramos de CO y un coste de electricidad de menos de 0.4 €. Estos costes e impactos ambientales mínimos son despreciables con respecto a los recursos necesarios para entrenar modelos a gran escala como GPT-4, que requieren una energía, emisiones e inversión financiera mucho mayores.




Matriz de Confusión - Conjunto de Validación

En cuanto a los resultados del algoritmo de clasificación, debido a la complejidad del conjunto de datos dada la presencia de animales camuflados y la variedad de especies, el modelo de clasificación fue entrenado durante 10 épocas pero se detuvo en la 6ª época debido al uso de un 'early stopping', ya que el modelo no mejoraba pasado este punto. La matriz de confusión de los datos de entrenamiento mostró que las imágenes de las

clases "Amphibian" y "Flying" fueron frecuentemente mal clasificadas como "Terrestrial", reflejando similitudes ambientales. Además, el 81% de las imágenes de la clase "Other" fueron mal clasificadas como "Terrestrial", probablemente debido a una categorización incorrecta por parte del creador del conjunto de datos. Los resultados del conjunto de validación otorgaron peores resultados: solo el 36% de las imágenes de "Amphibian" fueron clasificadas correctamente, con un 42% etiquetadas incorrectamente como "Terrestrial", y todas las imágenes en la categoría "Other" fueron mal clasificadas. El modelo alcanzó una precisión de validación del 66%, que podría mejorarse potencialmente eliminando o redistribuyendo mejor las imágenes de la clase "Other".

| Modelo | Número de parámetros | Conjunto de datos | Tiempo de Entrenamiento (h:min:s) | Número de épocas |
|---|---|---|---|---|
| UNET | 7,771,873 | D_NONCAM | 06:57:34 | 12 |
| | | D_BOTH | 01:04:44 | 12 |
| Attention | 9,344,841 | D_NONCAM | 17:15:33 | 20 |
| | | D_BOTH | 02:41:39 | 20 |
| Residual | 9,787,497 | D_NONCAM | 18:50:35 | 20 |
| | | D_BOTH | 03:04:55 | 20 |

Los modelos de segmentación (U-Net, Attention U-Net y R2 Attention U-Net) fueron entrenados en dos conjuntos de datos: $D_{NONCAM}$ (imágenes no camufladas) y $D_{BOTH}$ (imágenes mixtas). Los tiempos de entrenamiento variaron significativamente debido al tamaño de los conjuntos de datos y a la complejidad de los modelos. Por ejemplo, Attention U-Net y R2 Attention U-Net, con parámetros y complejidad adicionales, tardaron más en entrenarse en comparación con el U-Net básico. El entrenamiento de U-Net se detuvo prematuramente debido a la falta de mejora en la pérdida de validación. Las curvas de pérdida de entrenamiento y validación para U-Net mostraron un aprendizaje efectivo pero indicaron 'overfitting' hacia el final. Attention U-Net y R2 Attention U-Net demostraron una mejora consistente en la pérdida de validación, con Attention U-Net experimentando un aumento temporal en la pérdida probablemente debido a la naturaleza estocástica del optimizador Adam. Ambos modelos mostraron potencial para obtener mejores resultados con un entrenamiento prolongado.

| Modelo | Número de parámetros | Conjunto de Datos de Entrenamiento | Conjunto de Datos de Evaluación | IoU |
|---|---|---|---|---|
| **UNET** | 7,771,873 | D_NONCAM | CAM | 0.452 |
| | | | Conjunto de Entrenamiento | 0.866 |
| | | | Conjunto de Validación | 0.832 |
| | | D_BOTH | Conjunto de Entrenamiento | 0.646 |
| | | | Conjunto de Validación | 0.637 |
| **Attention** | 9,344,841 | D_NONCAM | CAM | 0.508 |
| | | | Conjunto de Entrenamiento | 0.865 |
| | | | Conjunto de Validación | 0.831 |
| | | D_BOTH | Conjunto de Entrenamiento | 0.592 |
| | | | Conjunto de Validación | 0.590 |
| **Residual** | 9,787,497 | D_NONCAM | CAM | 0.520 |
| | | | Conjunto de Entrenamiento | 0.808 |
| | | | Conjunto de Validación | 0.757 |
| | | D_BOTH | Conjunto de Entrenamiento | 0.559 |
| | | | Conjunto de Validación | 0.532 |

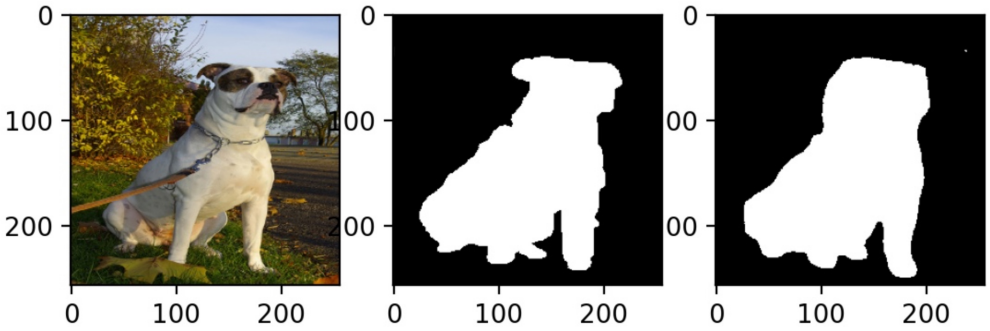
El rendimiento de la segmentación se evaluó utilizando 'Intersection over Union' (IoU) como métrica. Los modelos entrenados en $D_{NONCAM}$ y evaluados en el conjunto de datos CAM de animales camuflados mostraron:

- U-Net: IoU = 0.452
- Attention U-Net: IoU = 0.508
- R2 Attention U-Net: IoU = 0.520

Los modelos entrenados en $D_{BOTH}$ y evaluados en el conjunto de validación mostraron:

- U-Net: IoU = 0.637
- Attention U-Net: IoU = 0.590
- R2 Attention U-Net: IoU = 0.532

Entre los modelos entrenados en $D_{NONCAM}$, R2 Attention U-Net tuvo el IoU más alto para los animales camuflados, seguido por Attention U-Net y luego U-Net. Por el contrario, para los modelos entrenados en $D_{BOTH}$, U-Net fue el que mejor desempeño tuvo, seguido por Attention U-Net y R2 Attention U-Net. Estos resultados destacan la efectividad de los modelos en imágenes no camufladas, pero revelan desafíos en la segmentación de animales camuflados. Los problemas potenciales incluyen la complejidad de las imágenes camufladas, la insuficiencia de datos camuflados y las limitaciones de la arquitectura del modelo para manejar tales casos. A pesar de estos desafíos, modelos como R2 Attention U-Net mostraron capacidades de generalización prometedoras y el potencial para mejorar con entrenamiento adicional y ajustes en la estructura del modelo.




U-Net entrenado en $D_{NONCAM}$ - Segmentación de una imagen no camuflada del conjunto de validación (Imagen Original / Máscara / Predicción)

Los ejemplos visuales de los resultados de segmentación demuestran el rendimiento de los modelos. Para imágenes no camufladas, todos los modelos se desempeñaron bien, segmentando con precisión los animales. En cambio, la segmentación de imágenes camufladas reveló limitaciones, con modelos como U-Net luchando por distinguir el animal del fondo y R2 Attention U-Net mostrando potencial para mejorar en la forma de predicción y el reconocimiento de detalles. Estas observaciones subrayan la necesidad de capacidades de modelo mejoradas y datos camuflados adicionales para mejorar la precisión de la segmentación en escenarios complejos.

Como conclusión, la inteligencia artificial está evolucionando rápidamente como una tecnología transformadora con implicaciones significativas para el futuro. Este proyecto, que amplía un estudio realizado en colaboración con el Grupo CS, examina los impactos económicos y ambientales de la IA mientras aborda el desafío específico de la segmentación de animales camuflados para explorar las complejidades de desarrollar modelos de IA sofisticados. La investigación en varias arquitecturas de Redes Neuronales Convolucionales (CNN), como U-Net, Attention U-Net y R2 Attention U-Net, revela que, aunque estos modelos destacan en la segmentación de animales no camuflados, tienen dificultades con imágenes camufladas, lo que resalta la dificultad de diseñar sistemas de IA para escenarios complejos del mundo real. El modelo R2 Attention U-Net mostró potencial y sugiere que arquitecturas mejoradas y mayores recursos computacionales podrían mejorar el

rendimiento. Económicamente, se espera que la IA impulse la productividad y fomente el crecimiento, aunque también puede afectar los mercados laborales al cambiar la demanda, afectando la distribución de la desigualdad de ingresos. Además, el impacto ambiental de las crecientes necesidades computacionales de la IA plantea preocupaciones sobre el consumo de energía y las emisiones de carbono. Este proyecto subraya la necesidad de mejores metodologías para evaluar y mitigar los costes ambientales de la IA y destaca la importancia de equilibrar los beneficios de la tecnología con sus desafíos económicos y ambientales.

# 1 Introduction

In recent years, Artificial Intelligence (AI) has rapidly advanced, emerging as a leading force in technological innovation. It has sparked significant transformations across multiple industries and fundamentally reshaped our relationship with technology and the world. At its essence, AI endeavors to mimic and potentially exceed human cognitive functions using computational models and algorithms. This capacity empowers machines to interpret, comprehend, and respond to complex data, capabilities previously attributed exclusively to human intelligence.

Central to AI's capabilities is Computer Vision, a discipline within AI that focuses on enabling machines to interpret and understand visual information from the world. This field has seen remarkable advancements, particularly with the rise of deep learning techniques such as Convolutional Neural Networks (CNNs). These neural networks are inspired by the biological processes of the human brain and have proven exceptionally adept at tasks like image classification, object detection, and image segmentation.

Performing computer vision tasks in camouflaged environments presents a unique set of challenges due to the inherent complexity of detecting and identifying objects that blend seamlessly into their surroundings. Camouflage allows animals to evade predators or conceal themselves from prey, making them exceptionally difficult to detect even for human observers. For artificial intelligence systems, distinguishing camouflaged animals from their backgrounds requires not only advanced image processing techniques but also a deep understanding of context.

The intricacy of camouflaged environments offers a compelling area for further research in computer vision for several reasons. Firstly, developing effective algorithms to detect and segment camouflaged animals can have significant implications for ecological studies and wildlife conservation efforts.

Secondly, advancing computer vision capabilities in camouflaged environments can lead to innovations in military and defense applications. Enhancing the ability to detect concealed objects or individuals in natural settings is essential for improving situational awareness and operational effectiveness in military operations. These advancements can contribute to enhancing security measures and protecting personnel in challenging and dynamic environments.

Moreover, exploring the complexities of camouflaged environments through computer vision research can drive innovation in algorithm development and machine learning techniques. Tackling the challenges posed by camouflage requires advancements in image segmentation, pattern recognition, and contextual understanding within AI systems. These advancements not only benefit wildlife conservation and defense applications but also have broader implications for other fields such as medical imaging, remote sensing, and robotics.

In essence, investigating computer vision tasks in camouflaged environments represents a frontier of research that combines technological innovation with real-world applications in ecology, defense, and beyond. The investigation developed in this project is an extension of a study performed in collaboration with CS Group.

# 2 State of the Art

## 2.1 The Origin of modern AI and Neural Networks

The Perceptron and Threshold Logic Units (TLUs) [1] are foundational concepts in artificial intelligence, particularly in the development of neural networks. These models have been around for several decades, tracing their origins back to the mid-20th century. Despite their age, they have recently gained significant traction and widespread use due to their role as the building blocks for more complex neural network structures.

### 2.1.1 The Perceptron

The perceptron, introduced by Frank Rosenblatt in 1958 [2], is one of the simplest types of artificial neural networks. It is based on the Threshold Logic Unit where each input is associated with a weight:

$$z = w_1 * x_1 + w_2 * x_2 + ... + w_n * x_n = X^T * W$$

then a bias term is added and finally a step function is applied, outputting the result:

$$f_w(X) = step(z + b)$$

sometimes instead of the step function a sign function is used.

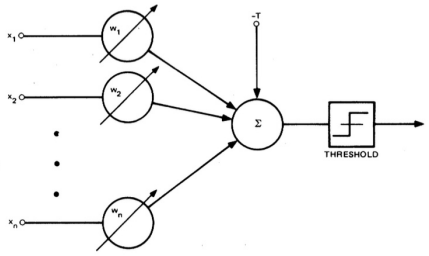



Figure 1: Diagram of a Threshold Logic Unit [citation]

A single TLU can be used for simple linear binary classification. It computes a linear combination of the inputs, if the output exceeds a threshold, it outputs the positive class. A perceptron is composed of a single layer of TLU's, with each TLU connected to all the inputs

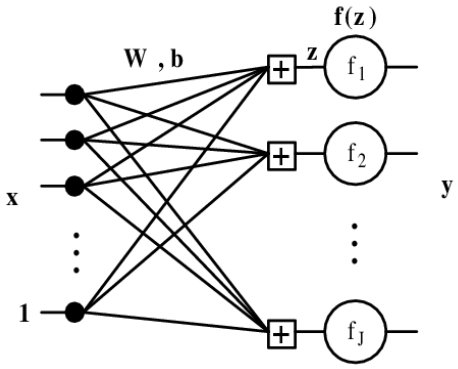



Figure 2: Perceptron (one layer neural network)

### 2.1.2 Perceptron Training

The perceptron is fed one training instance at a time, and for each instance it makes predictions. For every output neuron that produced a wrong prediction, it reinforces connection weights that would have contributed to the correct prediction.

$$w_{i,j}^{next} = w_{i,j} + \eta * (y_j - \hat{y_j}) * x_i$$

- $w_{i,j}$ is the connection weight between the $i^{th}$ input and the $j^{th}$ output
- $x_i$ is the $i^{th}$ input
- $\hat{y_j}$ is the output of the $j^{th}$ neuron
- $y_j$ is the target output of the $j^{th}$ output neuron
- $\eta$ is the learning rate

### 2.1.3 Multilayer Perceptron

The Multilayer Perceptron (MLP) is a class of feedforward artificial neural networks. It consists of at least three layers of nodes: an input layer, one or more hidden layers, and an output layer. Each layer is fully connected to the next one, meaning every node in one layer connects to every node in the subsequent layer. The primary purpose of an MLP is to map input data to the appropriate output through a series of linear transformations followed by non-linear activations.

- **Input Layer**: The input layer receives the raw input data. Each node in this layer corresponds to one feature in the input dataset.
- **Hidden Layers**: These layers perform most of the computation in an MLP. A hidden layer is composed of nodes (neurons) where each node applies a linear transformation to the input received from the previous layer and then applies a non-linear activation function, such as the sigmoid, tanh, or ReLU (Rectified Linear Unit). This non-linearity enables the network to capture complex patterns in the data.
- **Output Layer**: The output layer produces the final prediction. The type of activation function used in this layer depends on the nature of the task. For classification tasks, a softmax activation is typically used to output a probability distribution over classes. For regression tasks, a linear activation function might be used to produce continuous values.

## 2.2 Gradient Descent and Back-propagation: How do Machines Learn

The following definitions are based on the explanations from the book "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron [3].

### 2.2.1 Gradient Descent

GD is a generic optimization algorithm, its general idea is to tweak parameters iteratively in order to minimize a cost function. It measures the local gradient of an error function (that compares the desired output to the actual output of our system) with regard to the parameter vector $\Theta$, and then it modifies it in the direction of descending gradient. It starts by filling $\Theta$ with random values (random initialization). Then it improves it gradually, taking one baby step at a time, each step attempting to decrease the cost function, until the algorithm converges to a minimum.

The size of the steps is important, determined by the learning rate hyperparameter. If it is too small, the algorithm will have to go through many iterations to converge or it could converge to a local minimum. If it is too high the algorithm might diverge.

When the different features or parameters that compose $\Theta$ have very different scales ($\theta_1 >> \theta_2$) one parameter could take much longer to converge than the others since smaller $\theta_i$ needs greater change to affect the cost function. Therefore it is important to ensure that all features have similar scale when applying the algorithm.

#### 2.2.2 Batch gradient descent

To implement GD,it is necessary to compute the gradient of the cost function with respect to each model parameter $\theta_j$. This can be understood as calculating how much will the cost function change if $\theta_j$ is changed a little bit, i.e. a partial derivative must be computed.

$$\frac{d}{d\theta_j} MSE(\Theta) = \frac{2}{m} \sum_{i=1}^{m} (\Theta^T X^{(i)} - y^{(i)}) x_j^{(i)}$$

It is possible to compute all the partial derivatives in one go through the gradient vector $\nabla_\Theta MSE(\Theta)$

$$\nabla_\Theta MSE(\Theta) = \begin{bmatrix} \frac{d}{d\theta_0} MSE(\Theta) \\ ... \\ \frac{d}{d\theta_n} MSE(\Theta) \end{bmatrix} = \frac{2}{m} \mathbf{X}^T (\mathbf{X}\Theta - Y)$$

Notice how the formula involves calculations over the full training set $\mathbf{X}$ (all m instances) at each GD step. This means that the computation uses the whole batch of training data at each step, which makes it terribly slow for large training sets.

The gradient vector points uphill, therefore it is necessary to go on the opposite direction. This means subtracting $\nabla_\Theta MSE(\Theta)$ from $\Theta$. Now a multiplication between the gradient vector and the learning rate is made in order to determine the size of the downhill step .

$$\Theta^{next} = \Theta - \eta \nabla_\Theta MSE(\Theta)$$

#### 2.2.3 Stochastic Gradient Descent

Stochastic GD picks a random instance in the training set at every step and computes the gradients based only on that instance. Therefore it is much faster than Batch GD.

Nevertheless, given that it computes the gradient for a randomly chosen instance at a time, the gradient that reduces the error of a single instance might increase it for the whole data set. For this reason, this algorithm is less regular than BGD and the error will only decrease on average. Moreover, it will get closer to the minimum with time, but once it gets there it won't stop, it will continue to bounce around the optimum. That's why when the algorithm is stopped the values obtained will be good but not optimal. (When the cost function is very irregular, the bouncing around can also help scape local minima). One solution to settle at the minimum while escaping local minima is to gradually reduce the learning rate. The function that determines the learning rate is called the learning schedule. By convention the training is done over rounds of m iterations, where each round is called an epoch.

It is also very important to select the instances randomly (or to shuffle the instances at the beginning of each epoch) to ensure convergence towards a global optimum.

### 2.2.4 Mini-Batch Gradient Descent

The idea is to compute the gradients on small random sets of instances called Mini-Batches. It is less erratic than SGD but it may be harder for it to escape local minima.

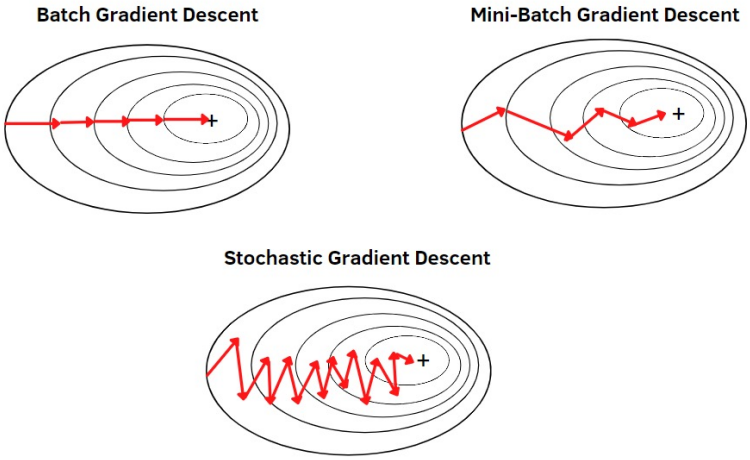



Figure 3: Gradient Descent variants

### 2.2.5 Backpropagation

In short, it is GD using an effective technique for computing the gradients automatically: in just two passes through the network (one forwards and one backwards), the propagation algorithm is able to compute the gradient of the network's error with regard to every single model parameter. In other words, it can find how each connection weight and each bias term should be tweaked in order to reduce the error. Once it has this gradients, it performs a regular GD step (aiming to reduce the loss function associated with the training) and the whole process is repeated until the network converges to the solution. To properly explain the algorithm a simple Neural Network model is presented in Figure 4:

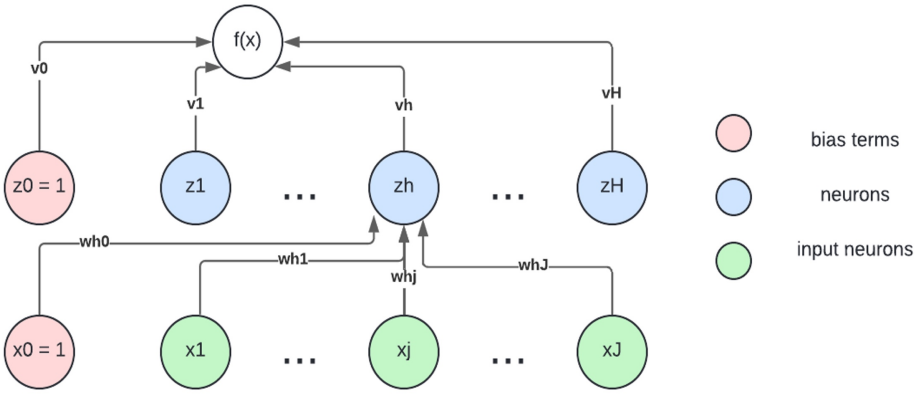



Figure 4: Simple NN model to explain backpropagation

To make sure the following equations (quite notation intensive are well understood, a description of the model and its notations is provided).

- $X = (x_0, ..., x_j, ..., x_J)^T$ represents the vector of inputs that is provided to the model. Notice how $x_0 = 1$ since $w_{h0} \cdot x_0$ is the bias term of the intermediate neuron h.
- $Y = (y^{(1)}, ..., y^{(i)}, ..., y^{(m)})^T$ represents the output vector expected from the model. In the configuration from Figure 4, given that there is only one output neuron, $m = 1$, but the equations will be generalized for $m \in N^*$.
- $F(X) = (f(X)^{(1)}, ..., f(X)^{(i)}, ..., f(X)^{(m)})^T$ represents the vector of predicted outputs by the model. Once again, in Figure 4 there is a single output neuron so $m = 1$.
- $V^{(i)} = (v_0^{(i)}, ..., v_h^{(i)}, ..., v_H^{(i)})^T$ is the weights vector associated to the output neuron i. $v_h^{(i)}$ is the weight value that links the output neuron i with the intermediate neuron h.
- $Z = (z_0, ..., z_h, ..., z_H)^T$ represents the output vector of the intermediate neuron h. Each neuron h is connected to all output neurons i through $v_h^{(i)}$. Notice how $z_0 = 1$ since $v_0^{(i)} \cdot z_0$ is the bias term of the output neuron i.
- $W_h = (w_{h1}, ..., w_{hj}, ..., w_{hJ})^T$ represents the weight vector that connects each intermediate neuron h with all the input neurons j.
- **Forward pass**: each instance is passed to the network's input layer, which sends it to the first hidden layer where the output of all neurons of this layer is computed. The result is passed on to the next layer, its output is computed and passed to the next layer, and so on until we get to the last layer, the output layer ($f(x)$). It is like making predictions, except all intermediate results are preserved since they are need for the backward pass.

$$z_h = \phi(W_h^T X)$$

$$f(X)^{(i)} = V^{(i)T} \cdot Z = v_0^{(i)} + \sum_{h=1}^{H} v_h^{(i)} \cdot z_h$$

  Next the algorithm measures the network's output error (i.e. it uses a cost function that compares the desired output and the actual output of the network, and returns some measure of the error).

$$Error(f(X), Y) = \frac{1}{2m} \sum_{i=1}^{m} (f(X)^{(i)} - y^{(i)})^2$$

  or:

$$Error^{(i)}(f(X)^{(i)}, y^{(i)}) = \frac{1}{2}(f(X)^{(i)} - y^{(i)})^2$$

- **Backward pass**: then it computes how much each output connection contributed to the error. This is done analytically by applying the chain rule to compute the gradient. It measures how much of these error contributions came from each layer below, working backwards until the algorithm reaches the input layer. This reverse pass efficiently measures the error gradient across all the connection weights in the network by propagating the error gradient backwards through the network.

  In general $\Delta\theta = -\eta \frac{dError^{(i)}(f(X)^{(i)}, y^{(i)})}{d\theta}$

$$\Delta v_h^{(i)} = \eta_1 (f(X)^{(i)}) - y^{(i)}) z_h$$

$$\Delta w_{hj} = -\eta \frac{dE^{(i)}}{dw_{h,j}} = -\eta \frac{dE^{(i)}}{df(X)^{(i)}} \cdot \frac{df(X)^{(i)}}{dz_h} \cdot \frac{dz_h}{dw_{h,j}} = \eta(y^{(i)} - f(X)^{(i)}) v_h^{(i)} \phi'(W_h^T X) x_j$$

- **GD step**: Finally, the algorithm performs a GD step to tweak all the connection weights on the network, using the error gradients it just computed.
- **Backpropagation algorithm**: At first, initialize all $v_h^{(i)}$ and $w_{hj}$ randomly. Then, repeat until convergence:

**for** $i = 1$ **to** $m$ **do**
  $f(X)^{(i)} = V^{(i)T} Z$
  **for** $h = 1$ **to** $H$ **do**
    $z_h = \Phi(W_h^T \cdot X)$
    $\Delta v_h^{(i)} = \eta_1 (Y^{(i)} - f(X)^{(i)}) z_h$
    $v_h^{(i)} \leftarrow v_h^{(i)} + \Delta v_h^{(i)}$
    **for** $j = 1$ **to** $J$ **do**
      $\Delta w_{hj} = \eta (Y^{(i)} - f(X)^{(i)}) v_h^{(i)} \phi'(W_h^T X) x_j$
      $w_{hj} \leftarrow w_{hj} + \Delta w_{hj}$
    **end for**
  **end for**

#### 2.2.6 Activation Functions

Activation functions are needed for the following reason: If several linear transformations are chained, the result is still a linear transformation. For example: $f(x) = 2x + 3$ and $g(x) = 5x - 1$, chaining these two linear function gives another linear function: $f((g(x))) = 2(5x - 1) + 3 = 10x + 1$. So if some non-linearity is not added between layers, even a deep stack of layers is equivalent to a single one. It's not possible to solve complex problems with that, conversely a Deep Neural Network (DNN) can, in theory, approximate any continuous function (if large enough).

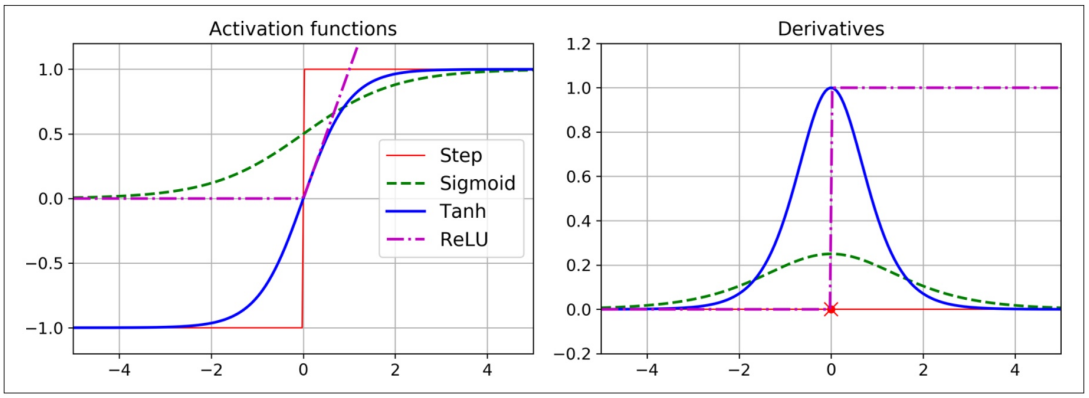



Figure 5: Some activation functions and their derivatives

- **Logistic (sigmoid) function**:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

This function replaced the step function to allow the backpropagation algorithm to work properly. Since the step function contains only flat segments, there is no gradient to work with. While the sigmoid function has a well-defined nonzero derivative everywhere.

- **Hyperbolic tangent function**:

$$tanh(z) = 2\sigma(2z) - 1$$

It is S-shaped, continuous and differentiable, its output values range from -1 to 1. This range tends to make each layer's output more or less centered around 0 at the beginning of training, which often speeds up convergence.

#### 2.2.7 Non Saturating Activation Functions

Activation functions often lead to the vanishing gradient problem (explained in depth later on), where gradients diminish during backpropagation, making the training of deep networks challenging. The advent of non-saturating activation functions marked a significant breakthrough, addressing some of these limitations and improving training efficiency and accuracy.

- **ReLU (Rectified Linear Unit)**

The Rectified Linear Unit (ReLU) function is defined as:

$$\text{ReLU}(z) = \max(0, z)$$

ReLU became popular due to its simplicity and efficiency. It does not saturate for positive inputs, allowing gradients to flow through the network effectively, which mitigates the vanishing gradient problem. However, ReLU is not without its drawbacks. It suffers from the "dying ReLU" problem, where neurons can permanently deactivate if their inputs are negative, leading to a large portion of the network being inactive.

- **Leaky ReLU**

To address the dying ReLU problem, a variant called Leaky ReLU was introduced. The Leaky ReLU function is defined as:

$$\text{LeakyReLU}_\alpha(z) = \max(\alpha z, z)$$

where $\alpha$ is a small positive constant, typically set to 0.01.

Leaky ReLU allows a small, non-zero gradient when the input is negative, ensuring that neurons do not become inactive. A 2015 paper [5] explored different values of $\alpha$ and found that a larger $\alpha$ (e.g., 0.2) can improve performance, though the optimal value depends on the specific task and dataset.

- **Randomized Leaky ReLU (RReLU)**

Randomized Leaky ReLU (RReLU) is a variation where $\alpha$ is randomly selected from a given range during training and is fixed during testing. This stochastic approach acts as a regularizer, reducing the risk of overfitting, and has been shown to perform well in practice.

- **Parametric ReLU (PReLU)**

  Parametric ReLU (PReLU) takes the concept of Leaky ReLU further by allowing the parameter $\alpha$ to be learned during training instead of being fixed. This adaptability gives PReLU more flexibility, leading to improved performance in some cases, especially on large datasets.

- **Exponential Linear Unit (ELU)**

  The Exponential Linear Unit (ELU) was proposed in a 2015 paper by Djork-Arné Clevert et al [6]. The ELU function is defined as:

$$\text{ELU}_\alpha(z) = \begin{cases} \alpha(\exp(z) - 1) & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases}$$

  ELU has several advantages over ReLU and its variants:

  - It takes on negative values for negative inputs, which helps the network converge to zero mean outputs and mitigates the vanishing gradient problem.
  - It has a smooth gradient for $z < 0$, reducing the likelihood of dead neurons.
  - It outperforms ReLU variants in terms of speed and accuracy on various tasks.

  However, ELU is slower to compute than ReLU due to its use of the exponential function.

- **Scaled Exponential Linear Unit (SELU)**

  Introduced in a 2017 paper by Günter Klambauer et al. [7], the Scaled Exponential Linear Unit (SELU) is a scaled version of the ELU. SELU is defined as:

$$\text{SELU}(z) = \lambda \text{ELU}_\alpha(z)$$

  where $\lambda$ is a scaling parameter.

  SELU is designed to ensure self-normalizing properties within a neural network, meaning that the outputs of each layer have a mean of 0 and a standard deviation of 1. This self-normalization reduces the vanishing/exploding gradient problems and often results in better performance than other activation functions, particularly in deep networks.

Choosing the right activation function is crucial for the performance of deep neural networks. While ReLU has been the go-to activation function for many applications, its variants like Leaky ReLU, PReLU, and ELU offer improvements by addressing specific issues such as the dying ReLU problem and vanishing gradients. The choice between these functions depends on the specific requirements of the task at hand, the architecture of the network, and the dataset used. As deep learning continues to evolve, so too will the innovations in activation functions, driving further advancements in this field.

## 2.3 Some NN problems

### 2.3.1 Vanishing Gradients Problem

Remember that the backpropagation algorithm propagates the error gradient from the output layer to the input layer. Unfortunately, gradients often get smaller and smaller

as the algorithm progresses down to the lower layers. Since given that the computation of the gradient follows the chain rule, if an intermediate derivative is close to zero, the following ones will be highly attenuated. As a result, the GD update leaves the lower layer's connections virtually unchanged. We call this the Vanishing Gradient Problem.

In some cases the opposite can happen: the gradients grow bigger and bigger until layers get insanely large weight updates and the algorithm diverges. It is the Exploding Gradients Problem, which emerges in Recurrent Neural Networks.

It has been argued that the Vanishing Gradient Problem comes or is aggravated by two factors: the use of the logistic sigmoid function for activation and a specific weight initialization technique.

- **Sigmoid function**: when the inputs become very large (negative or positive), the function saturates at 0 or 1, with a derivative extremely close to 0. Thus, when backpropagation kicks in, it has almost no gradient to propagate back, and it keeps getting diluted as it progresses down. Therefore there is nothing left for the lower layers.
- **Initialization scheme**: If we use the sigmoid and initialize our weights with a normal distribution with mean 0 and standard deviation of 1, we can prove that the variance of the outputs of each layer is much greater than the variance of the inputs. As we go forward in the network, this often leads to saturation of the activation function and therefore very small gradients.
- **Solve VGP**: To ensure that when flowing through the network the signal doesn't die out or explode we need the variance of the outputs of each layer to be equal to the variance of the inputs and we need the variance of the gradients to have equal variance before and after flowing through a layer in reverse direction. It is not possible unless the layer has an equal number of inputs and neurons (these numbers are called $fan_{in}$ and $fan_{out}$ of the layer). An effective initialization method has been found depending on the activation function:

  Given that: $fan_{avg} = \frac{(fan_{in}+fan_{out})}{2}$

| INIT. | ACTIVATION FUNC. | $\sigma^2$(NORMAL) |
|---|---|---|
| Glorot | None, tanh, logistic, softmax | $\frac{1}{\text{fan\_avg}}$ |
| He | ReLu and variants | $\frac{2}{\text{fan\_in}}$ |
| LeCun | SELU | $\frac{1}{\text{fan\_in}}$ |

  For all activation functions we can also use a uniform distribution between $r$ and $-r$ where $r = \sqrt{3\sigma^2}$.

  Another useful tool that can help mitigate this problem is Batch Normalization. **Batch Normalization** can significantly help reduce the vanishing gradients problem by stabilizing the distribution of inputs to each layer during training. By normalizing the inputs of each layer to have a mean of zero and a standard deviation of

one, batch normalization prevents the activation functions, like the sigmoid, from saturating. This means that the gradients passed back during backpropagation are less likely to become extremely small, which helps to maintain a more consistent gradient flow through the network. Additionally, batch normalization reduces the dependency on careful weight initialization, as it helps maintain appropriate variance across layers, further mitigating the risk of gradients vanishing. This stabilization not only speeds up training but also makes the network less sensitive to the choice of hyperparameters, leading to more robust and efficient learning.

### 2.3.2 Overfitting

Overfitting is a common issue in machine learning, particularly when training complex models like deep neural networks. It occurs when a model learns the training data too well, capturing not only the underlying patterns but also the noise and outliers in the data. As a result, while the model performs exceptionally well on the training data, it fails to generalize to new, unseen data, leading to poor performance on validation or test sets. Overfitting is typically a sign that the model has too much capacity relative to the amount of training data, allowing it to memorize the data rather than learning generalizable features.
To combat overfitting, several regularization techniques can be applied, two of the most effective being early stopping and dropout.

- **Early stopping** is a technique that monitors the model's performance on a validation set during training. As training progresses, the model's performance on the validation set is expected to improve, but once overfitting begins, this performance typically starts to degrade. Early stopping halts the training process as soon as the validation error starts to increase, thereby preventing the model from further overfitting to the training data.

- **Dropout** is another powerful technique used to prevent overfitting, particularly in deep neural networks. Dropout works by randomly "dropping out" (i.e., setting to zero) a fraction of the neurons in a layer during each training iteration. This forces the network to learn redundant representations of features, making it less reliant on any single neuron and therefore more robust. Dropout effectively creates a form of ensemble learning, where multiple different sub-networks are trained simultaneously, which improves the model's ability to generalize to new data. By using these techniques, deep learning practitioners can significantly reduce the risk of overfitting, leading to models that are more capable of performing well on unseen data.

## 2.4 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a specialized artificial neural network inspired by the human visual cortex's method of image processing. Similar to how humans first identify fundamental features such as edges and colors in an image, and then progressively analyze finer details and their interactions, CNNs replicate this approach by extracting features in a hierarchical fashion.

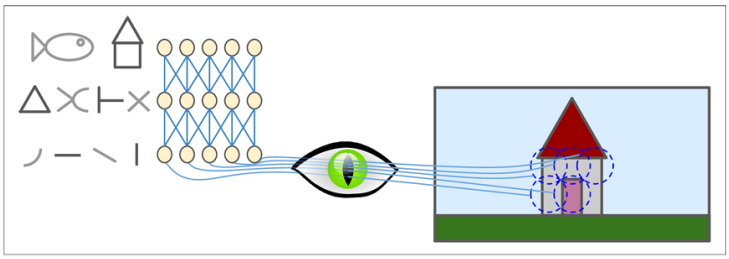


Figure 6: Hierarchical extraction of features by the human eye

The CNN are composed of multiple layers, each one of them applying a different filter. Those layers include :

- Fully connected layers ('dense layer')
- Convolutional layers
- Pooling layers

In a fully connected layer, every neuron establishes connections with all neurons in the preceding layer. These interconnected links form a network of weighted interactions, enabling the neural network to adeptly integrate features extracted from diverse layers. This characteristic renders fully connected layers particularly suitable for tasks demanding advanced reasoning and classification based on these integrated features.

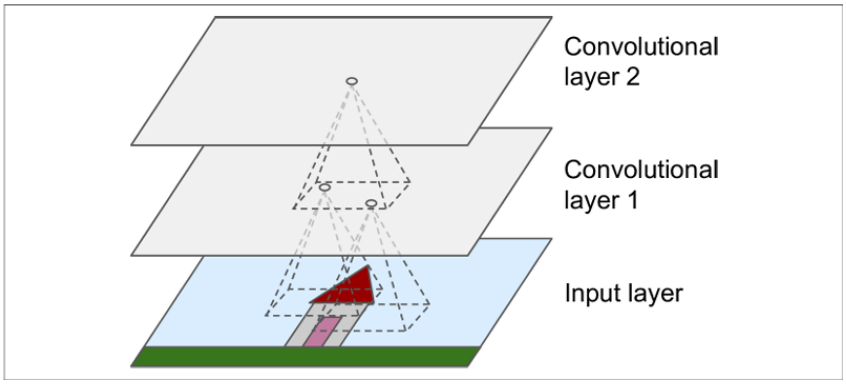



Figure 7: CNN layers with rectangular local receptive fields

The convolutional layers serve as the cornerstone of a CNN. Neurons in the initial convolutional layer are linked exclusively to pixels within their respective receptive fields, rather than every pixel in the input image. Similarly, neurons in subsequent convolutional layers connect solely to neurons positioned within small rectangles of the preceding layer (receptive fields). This hierarchical arrangement enables the network to focus initially on minute, low-level features in the initial hidden layer, progressively integrating them into more complex, higher-level features in subsequent layers.

The weights of a neuron can be visualized as a compact image matching the size of its receptive field. Each specific set of these weights is referred to as a filter or convolution kernel. For instance, consider a 7x7 matrix where all elements are 0 except for a central vertical line filled with 1's. This filter effectively emphasizes only the central vertical line within its receptive field while disregarding other details.

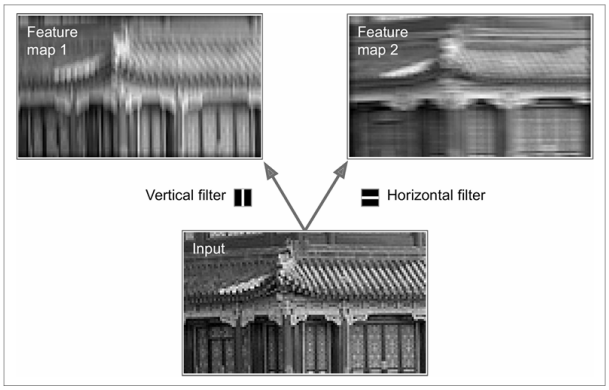



Figure 8: CNN layers with rectangular local receptive fields

When all neurons in a layer employ identical filters (along with the same bias term), the layer produces a feature map. This map accentuates the regions in an image that most strongly activate the filter. Throughout training, the convolutional layer autonomously learns the most effective filters for its specific task. Subsequently, the layers above it learn to combine these filters into more complex patterns.

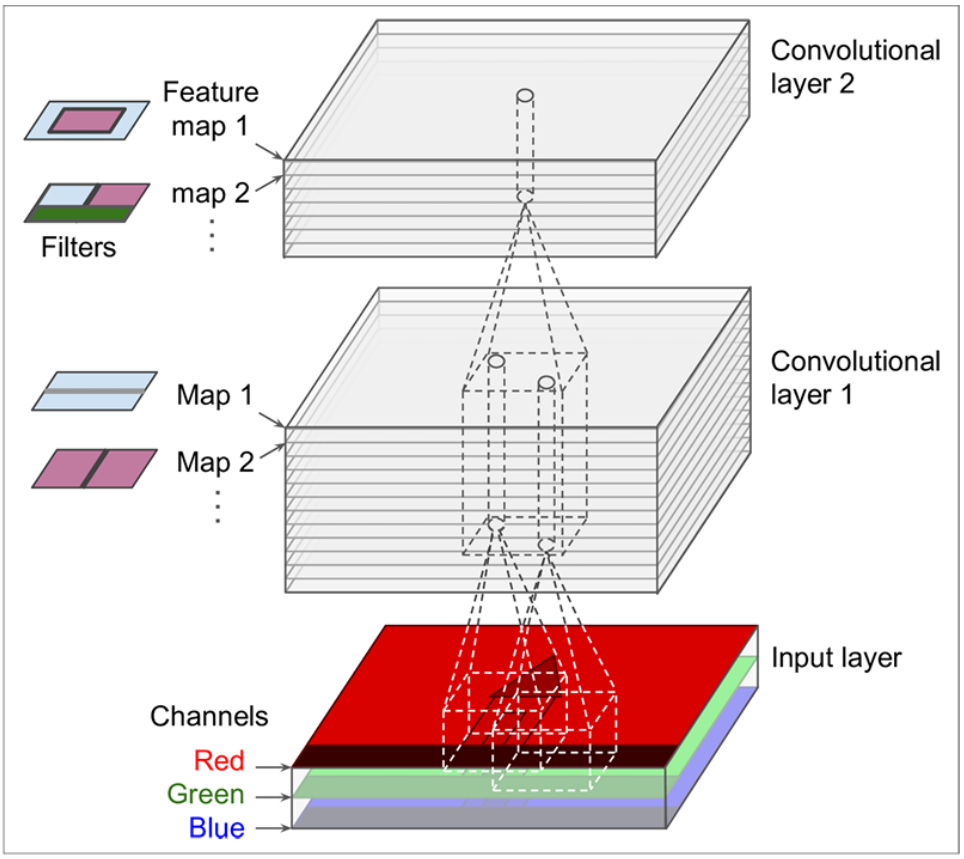



Figure 9: CNN layers with rectangular local receptive fields

A convolutional layer is best visualized in 3D because each layer contains multiple filters (the number can be chosen), resulting in one feature map generated per filter. Within each feature map, there is one neuron per pixel, and all neurons within the same feature map share identical parameters. Neurons across different feature maps, however, utilize distinct parameters. Each neuron's receptive field spans across all feature maps of the preceding layer, as described earlier. Essentially, a convolutional layer applies multiple trainable filters simultaneously to its inputs, allowing it to detect various features across the input space. In contrast with a Dense Neural Network (DNN), the fact that all neurons in a feature share the same parameters dramatically reduces the number of parameters in the model, and once the CNN has recognized certain pattern in one location, it can recognize it anywhere else.
Computing the output of a neuron in a convolutional layer:

$$z_{i,j,k} = b_k \sum_{u=0}^{f_h-1} \sum_{\nu=0}^{f_w-1} \sum_{k'=0}^{f_{n'}-1} x_{i',j',k'} w_{u,\nu,k',k}$$

with,

$$\begin{cases} i' = i * s_h + u \\ j' = j * s_w + \nu \end{cases}$$

In this equation:

- $z_{i,j,k}$ is the output of the neuron located in row i, column j in feature map k of the convolutional layer (layer l).
- As explained earlier, $s_h$ and $s_w$ are the vertical and horizontal strides, $f_h$ and $f_w$ are the height and width of the receptive field, and $f_n$ is the number of feature maps in the previous layer (layer l – 1).
- $x_{i,j,k}$ is the output of the neuron located in layer l – 1, row $i$, column $j$, feature map $k$ (or channel $k$ if the previous layer is the input layer).
- $b_k$ is the bias term for feature map k (in layer l). You can think of it as a knob that tweaks the overall brightness of the feature map k.
- $w_{u,v,k,k}$ is the connection weight between any neuron in feature map k of the layer l and its input located at row u, column v (relative to the neuron's receptive field), and feature map k.

The purpose of pooling layers is to subsample the input image to reduce computational load, memory usage, and the number of parameters, thereby decreasing the risk of overfitting. Similar to convolutional layers, each neuron in a pooling layer connects to a limited number of neurons in the previous layer within a small rectangular receptive field. However, pooling neurons do not have weights; they simply aggregate the inputs using an aggregation function like max or mean. The aim is to retain as much relevant information as possible while reducing the image size.

Moreover, a max pooling layer can introduce some level of invariance to small translations, a small amount of rotational invariance and a slight scale invariance. Such invariance can be useful in cases where the prediction should not depend on those details, such as classification tasks.

## 2.5 MirrorNet

MirrorNet, developped by Jinnan Yan et al. on the 2021 paper "MirrorNet: Bio-Inspired Camouflaged Object Segmentation" [8] is a bio-inspired framework which significantly improves the detection and segmentation of camouflaged objects in images. The framework is highlighted for its dual-stream approach, which processes both original and horizontally flipped images to better identify objects camouflaged within their environments.

### 2.5.1 Bio-Inspired Approach:

MirrorNet introduces a dual-stream network that includes a main stream for processing the original image and a mirror stream for the flipped image. This approach is inspired by natural systems and is designed to detect disruptions in environmental continuity, aiding in the detection of camouflaged objects. To properly understand why image flipping might be relevant in the detection of camouflaged objects in complex environments it is necessary to discusses the theoretical foundations underpinning the approach to detecting camouflaged objects, focusing on the distinction between viewpoint-invariant and viewpoint-dependent theories in biological vision studies:

- **Viewpoint-Invariant Theories**: These theories suggest that once an object has been visually encoded by the brain, it can be recognized from new angles without any loss in recognition accuracy, as long as the necessary features of the object are visible from those angles. This theory assumes a stable recognition capability across different viewpoints.
- **Viewpoint-Dependent Theories**: Contrary to the viewpoint-invariant perspective, these theories posit that recognition can degrade when viewing objects from unfamiliar angles. Recognition is more reliable when objects are seen from previously encountered or typical viewpoints.

The document points out that these theories were initially developed based on observations of non-camouflaged objects against clear, unambiguous backgrounds, such as a plain white background. This context is critical because non-camouflaged objects typically stand out against their backgrounds, making them easy to detect and recognize.

However, the challenge significantly increases with camouflaged objects, which blend into their backgrounds. Such objects share visual properties—like color and texture—with their surroundings, making them hard to distinguish. This blending diminishes the visual differences that are typically relied upon for object recognition.

The authors then describe an experiment leveraging a deep learning model, Fully Convolutional Network (FCN) [9], which computes the visual difference between camouflaged and non-camouflaged objects against their backgrounds. By analyzing these differences, they aim to understand how camouflaged objects can be better detected:

- The document mentions computing "difference distances" using the model pre-trained on the PASCAL-VOC [10] dataset, focusing on measuring differences in color and texture (using measures like RGB and l color spaces, and texton [11] for texture)
- Importantly, when images are flipped horizontally, the usual camouflage patterns are disrupted, making it easier to detect discrepancies between the camouflaged

objects and their backgrounds. This suggests that flipping images could be a practical approach to enhance the detection of camouflaged objects in automated systems.

- The analysis showed that while the color space differences were minimal, the textural and feature-based differences (computed from deep learning models) were more significant. This indicates that deep learning models, which can learn and identify complex patterns and textures, may be more effective at detecting camouflaged objects than simple color-based analyses.

Therefore the research paper suggests that a combination of innovative image manipulation techniques (like image flipping) and advanced deep learning models can provide a more robust framework for identifying objects that blend into their backgrounds. This dual approach—incorporating both intrinsic properties of the object and extrinsic manipulations of the image—enhances the detection capabilities of the system.

#### 2.5.2 Network Architecture:

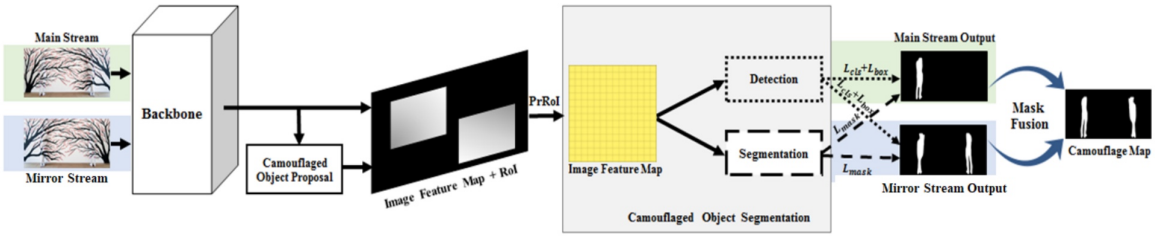



Figure 10: MirrorNet framework

Figure 10 depicts the structure of MirrorNet. As previously stated MirrorNet incorporates two streams, the primary stream handles segmentation of the original image, while the mirror stream manages segmentation of the horizontally flipped image, utilizing this bio-inspired strategy to disrupt the typical camouflage patterns in nature. Both streams proceed through stages of camouflaged object proposal and segmentation, resulting in the creation of segmentation masks. These masks from the two streams are then integrated to form a highly precise camouflage map at the pixel level. Here's a breakdown of each component:

- **Camouflaged object proposal**: The Camouflaged Object Proposal component is crucial for localizing camouflaged objects within the image. It uses the Region Proposal Network (RPN) [12] to predict potential object locations by outputting bounding boxes (Regions of Interest - RoI) of various sizes. Each RoI is processed to pool a fixed-size feature map from the broader image feature map, using techniques to maintain the alignment and accuracy of the features extracted relative to their position in the input image.
- **Camouflaged Object Segmentation**: This process involves using a multi-task loss function that combines:
  - Classification Loss ($\mathcal{L}_{\text{cls}}$): A cross-entropy loss that assesses the accuracy of object classification within the RoI.
  - Localization Loss ($\mathcal{L}_{\text{loc}}$): A Smooth L1 loss used to refine the position and size of the bounding box relative to the ground-truth.

- Mask Loss ($\mathcal{L}_{\text{mask}}$): Another cross-entropy loss, calculated specifically over the segmentation mask to ensure that each pixel within the RoI is correctly labeled as part of a camouflaged object or not.

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{loc}} + \mathcal{L}_{\text{mask}},$$

- **Mask Fusion**: To finalize the segmentation process, MirrorNet employs a "winner take all" strategy where bounding boxes from both streams are compared:
  - Bounding boxes with overlapping predictions are assessed, and those with lower confidence scores are discarded if they overlap significantly with higher score boxes.
  - The remaining masks are then combined, and their outputs are normalized to create the final, unified camouflage map that consistently covers the detected camouflaged objects across the entire image.
- **Backbone**: To perform the the Camouflaged Object Proposal and Object Segmentation they proceed to do a fine-tuning over some consolidated State-of-the-Arts such as ResNet [13] and ResNeXt. Where they trained the network for 120k iterations with the base learning rate of 0.00125, which is decreased by 10 times every time they reach the next steps at 80k iterations and 100k iterations.

#### 2.5.3 Experimental Validation:

The experiments were conducted using the CAMO dataset, created in the paper "Anabranch Network for Camouflaged Object Segmentation" [14], which includes images of camouflaged and non-camouflaged objects. The dataset is specifically designed to challenge segmentation models because it features objects that blend seamlessly into their backgrounds. The evaluation metrics used were F-measure [15], Intersection Over Union (IOU) [16], and Mean Absolute Error (MAE), providing a robust framework for assessing precision, recall, and error rates in segmentation tasks.

MirrorNet's performance was evaluated against several state-of-the-art models, including variants of ANet and FCN, which had been fine-tuned for the camouflaged object segmentation task. These models were assessed on their ability to distinguish between camouflaged and non-camouflaged objects in various settings.

- **Overall Performance**: MirrorNet consistently outperformed existing methods across all metrics. The results highlighted the advantage of using a dual-stream approach (original and mirror images) in detecting objects that are naturally designed to evade visual detection. The metrics clearly showed that MirrorNet, especially the variants using ResNeXt backbones, significantly outperformed all other models, obtaining an astonishing 82.2% of accuracy on IOU in the best case, confirming the effectiveness of the architectural choices and training strategies employed.
- **Effectiveness of Dual Streams**: The experiments demonstrated that the inclusion of a mirror stream, processing flipped images, significantly enhances the model's ability to detect discrepancies in texture and outline that are not apparent in the original orientation. This bio-inspired strategy effectively counters the camouflage techniques employed by the objects in the dataset.

# 3 Objectives and Problems to be Addressed

This project aims to explore and elucidate the multifaceted impacts of artificial intelligence (AI) on the economy, society, and the environment, with a particular emphasis on the development and deployment of highly performant AI models. Specifically, the project addresses two primary objectives: understanding the challenges in AI model development, particularly for complex tasks, and analyzing the broader implications of AI in terms of energy consumption, computational demands, and economic impact.

## 3.1 Objective 1: Understanding the Challenges in AI Model Development

One of the key objectives of this project is to delve into the technical complexities associated with developing convolutional neural networks (CNNs) for segmenting camouflaged animals in rural environments. Camouflaged objects present a significant challenge for conventional segmentation models due to their complex textures and minimal differences from their surroundings. By developing and evaluating several CNN-based models, including variations of the U-Net architecture, this project aims to uncover the specific difficulties encountered in this task. These insights will help identify potential solutions that are computationally efficient and cost-effective, thereby contributing to the field of computer vision and advancing the capabilities of AI in complex real-world scenarios.

## 3.2 Objective 2: Analyzing the Economic, Energetic, and Environmental Impacts

The second major objective of this project is to critically assess the economic, energetic, and environmental impacts of training large-scale AI models. Training heavy AI models, particularly those with millions of parameters, is an intensive process that requires significant computational resources. This process incurs substantial energetic costs due to the electricity required to power the GPUs and specialized hardware involved, which, in turn, contributes to a considerable carbon footprint. AI, particularly in the form of large-scale models, represents a significant economic factor due to its influence on various cost structures within companies and across industries. Training and maintaining AI models involves substantial investments in hardware, energy, and expertise, which can have profound effects on a company's financial performance.

## 3.3 Problems Addressed by the Project

This project addresses several critical problems associated with the current trajectory of AI development:

- **Technical Complexity and Scalability**: As AI models become more sophisticated, the technical complexity involved in their development increases. This project aims to provide insights into the specific challenges of developing AI models for complex tasks, such as segmenting camouflaged animals, which can inform strategies for optimizing these models without incurring excessive computational costs.
- **Energy Consumption and Environmental Impact**: The energy-intensive nature of training large AI models poses significant environmental concerns. This project seeks to highlight the energy consumption associated with AI development and propose more sustainable approaches, contributing to the ongoing discourse on the environmental impact of AI.

- **Economic Impact of AI**: The objective is to assess the economic impact of AI on society, with a particular emphasis on its effects on the labor market, including the displacement of jobs, changes in wage inequality, and shifts in demand for different skill levels. Additionally, the study will analyze the costs associated with training specific AI models to understand the broader economic implications of AI deployment
- **Benchmarking and Model Evaluation**: There is a need for a comprehensive understanding of how different AI models perform on specific tasks. This project addresses this by comparing the performance of various CNN architectures on the task of segmenting camouflaged animals, thereby contributing to the body of knowledge on model selection and optimization for specific use cases.

In summary, this project not only aims to push the boundaries of what AI can achieve in complex image segmentation tasks but also seeks to provide a holistic understanding of the broader implications of AI development. By doing so, it contributes to the ongoing efforts to make AI more sustainable, accessible, and effective in addressing real-world challenges.

# 4 Environmental constraints

### 4.0.1 Relevance of environmental impact

It's crucial to keep track of the environmental impact of AI, especially as these technologies continue to grow and become more widespread in various industries, driving both innovation and economic progress. AI systems, particularly those that use deep learning and large language models (LLMs), require a vast amount of computational power, leading to significant energy use. This high energy demand not only contributes to global warming by increasing the carbon footprint of Information and Communication Technologies (ICT) but also puts pressure on natural resources like water, which is essential for cooling data centers. As AI becomes more integrated into everyday operations, from personalized services to large-scale industrial processes, its overall environmental impact becomes a critical issue that must be carefully examined and addressed. Ignoring these environmental concerns could lead to unsustainable practices that worsen climate change, deplete natural resources, and ultimately counteract the benefits that AI is supposed to bring. As we move further into the digital era, it's up to researchers, developers, and policymakers to ensure that the growth of AI technologies is in line with environmental sustainability goals. This means not only understanding the energy and resource needs of AI but also finding ways to make these technologies more efficient and environmentally friendly. Monitoring the environmental impact of AI is, therefore, essential for creating a responsible technological ecosystem that balances progress with the well-being of our planet, ensuring that AI can be a positive force without harming the environment.

### 4.0.2 ICT's Impact on the Environment

Information and Communication Technology (ICT) encompasses a broad range of technologies that facilitate communication and the processing of information. This umbrella term includes various devices and systems such as radios, televisions, mobile phones, computers, and network hardware, as well as the software and applications that operate them. ICT also extends to the satellite systems that provide connectivity and the assorted services that rely on this infrastructure, from basic telephony to complex forms of distance

learning and videoconferencing. These technologies form the backbone of the modern information society, playing an essential role in both personal and professional spheres. As AI systems are embedded in software that runs on ICT hardware, and as they become increasingly vital in managing and processing large amounts of data, AI naturally fits within the broader definition of ICT.

The environmental impact of ICT is a topic of growing concern, particularly as the use of these technologies continues to expand. The study presented in this project represents a summary of the paper "Environmental Impact of Artificial Intelligence" [17]. These impacts can be analyzed through different orders, each reflecting a layer of interaction between ICT and the environment. Hilty et al. [18] have proposed a conceptual framework that classifies these effects into three orders or levels.

- The first order of ICT's environmental impact involves the direct effects of ICT technologies themselves. These effects are primarily associated with the physical existence and operation of ICT hardware, including the environmental consequences of producing, using, and disposing of devices like computers, mobile phones, and network infrastructure. The environmental impacts at this level are typically negative, encompassing the energy consumption required to operate these devices and the waste generated from their production and disposal. Additionally, the materials used in these technologies, such as rare earth metals and other non-renewable resources, contribute to environmental degradation through extraction and processing.
- The second order of environmental impact considers the application of ICT and how these technologies are optimized and substituted in various processes. While the direct use of ICT can have harmful environmental effects, its application can also lead to positive outcomes. For instance, ICT can improve efficiency in other sectors by optimizing energy use or substituting physical goods with digital alternatives, thereby reducing overall environmental impact. However, this level of impact also includes negative effects, such as the phenomenon of obsolescence, where technology becomes outdated quickly, leading to increased electronic waste. Additionally, the use of ICT can sometimes induce new forms of environmental harm, such as the energy-intensive nature of data centers required to support cloud computing services.
- The third order of ICT's environmental impact relates to systemic effects, which emerge over time as ICT drives structural and lifestyle changes in society. These changes can either mitigate or exacerbate environmental impacts. On the positive side, ICT can support transitions towards more sustainable practices, such as telecommuting, which reduces the need for transportation and consequently lowers greenhouse gas emissions. Conversely, ICT can also lead to rebound effects, where the efficiency gains achieved by ICT are offset by increased consumption elsewhere, ultimately negating the environmental benefits. This level of impact is complex and multifaceted, reflecting the deep entwinement of ICT with economic and social systems.

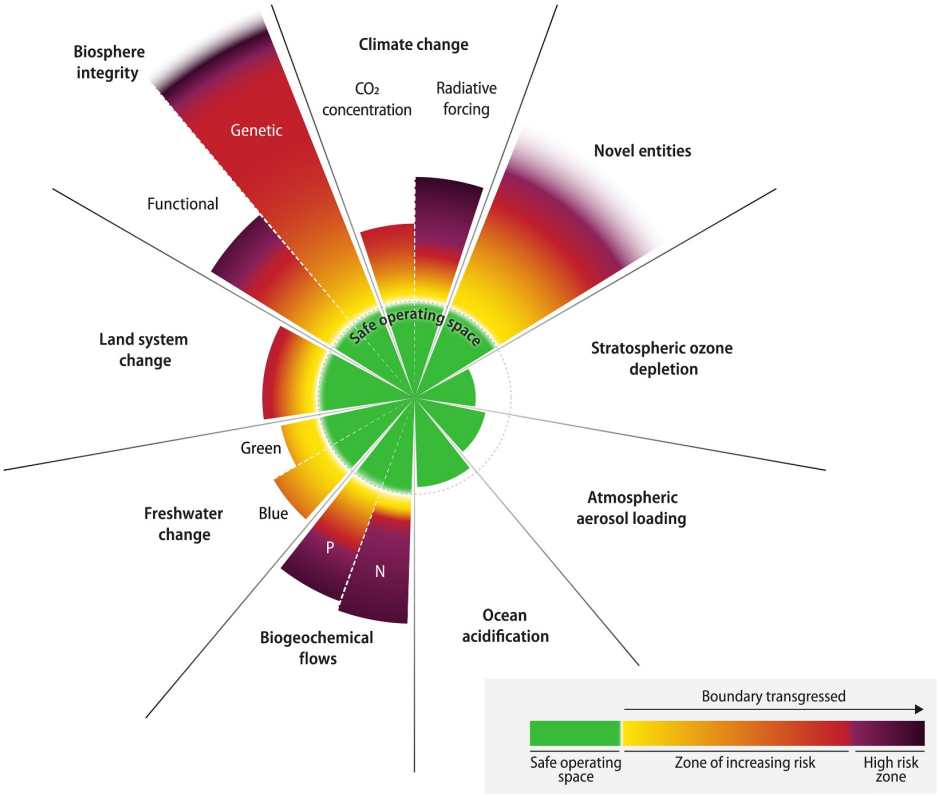



Figure 11: Current status of control variables for all nine planetary boundaries (Richardson et al. [c])

To evaluate these impacts holistically, a Multi-criteria Analysis of ICT's Environmental Impacts is crucial. The earth system, as described by the planetary boundaries concept introduced by Rockström et al. [19], is governed by nine key boundaries that define the environmental limits within which humanity can operate safely. These boundaries include climate change, biosphere integrity, land-system change, and freshwater use, among others. ICT's environmental impact must be assessed concerning these boundaries to understand the broader implications of technology on global sustainability. According to a study by Richardson et al. [20], from the nine boundaries, six have already been transgressed, signaling an urgent need to consider these limits in the development and deployment of ICT technologies. Figure 11 illustrates how various environmental impacts are distributed across these boundaries, with certain areas already approaching or exceeding safe limits.

In a detailed examination of ICT's environmental impact, Freitag et al.[21] reviewed post-2015 peer-reviewed papers to assess and compare the environmental effects of ICT, with a particular focus on its carbon footprint. They centered their analysis on the work of three research groups: Andrae and Elder, Belkir and Elmeligi, and Malmodin and Lundén. These groups utilized hybrid top-down and bottom-up approaches within Life Cycle Assessment (LCA) methodologies to estimate the greenhouse gas (GHG) emissions attributable to ICT by 2020. Based on these works, Freitag et al. have found predicted ICT emissions for 2020 between 1.0 and 1.7GtCO2e, which represent between 1.8% - 2.8% of worldwide GHG emissions for that year. The results obtained from each of the research groups are present in Figure 12.

| | Andrae and Elder | Belkhir and Elmeligi | Malmodin and Lundén |
|---|---|---|---|
| Estimation of ICT's impact for 2020 (% of global GHG emissions) | 1.5-6.3 | 1.9 - 2.3 | 1.9 |
| Year of the data used for predictions | 2011 | 2008 - 2011 | 2015 |
| Conflicts of interests | Working at Huawei | No obvious one | Working at Ericsson |
| Data transparency | Yes | Yes | No: non-reviewable confidential data, from ICT companies |

Figure 12: Comparison of Andrae Elder, Belkir Elmeligi and Malmodin Lundén works (Environmental Impacts of Artificial Intelligence [22])

The review also brought to light issues regarding transparency, particularly in the data used for these assessments. Some studies relied on confidential data that could not be independently verified, raising questions about the robustness of the conclusions. Furthermore, Freitag et al. suggested that these research groups might have underestimated the overall impact of ICT. They pointed out that the studies primarily focused on direct (first-order) emissions and did not adequately consider second and third-order effects, such as the indirect impacts of ICT on other sectors or the long-term systemic changes driven by ICT innovations.

One of the key critiques from Freitag et al. was the exclusion of rapidly growing ICT-related sectors, such as Artificial Intelligence (AI), the Internet of Things (IoT), and blockchain technologies, from the emissions estimates. These sectors are becoming increasingly significant within the ICT landscape and have the potential to substantially increase overall emissions. Additionally, the authors discussed the challenge of defining the boundaries of what constitutes ICT. Traditional elements like computers, smartphones, and data centers are typically included, but emerging technologies such as IoT and blockchain, which also rely heavily on ICT infrastructure, are often excluded despite their potential to significantly impact overall emissions.

Regarding the future of the sector, Andrae and Elder predict a significant rise in ICT-related greenhouse gas (GHG) emissions, with a potential 50% increase by 2030 and up to a 500% surge in the worst-case scenario. Belkir and Elmeligi share this concern, forecasting an exponential growth in emissions, potentially multiplying by 2.6 times by 2030 and 5.1 times by 2040. This projected growth is illustrated in Figure 13, which visualizes these potential increases.

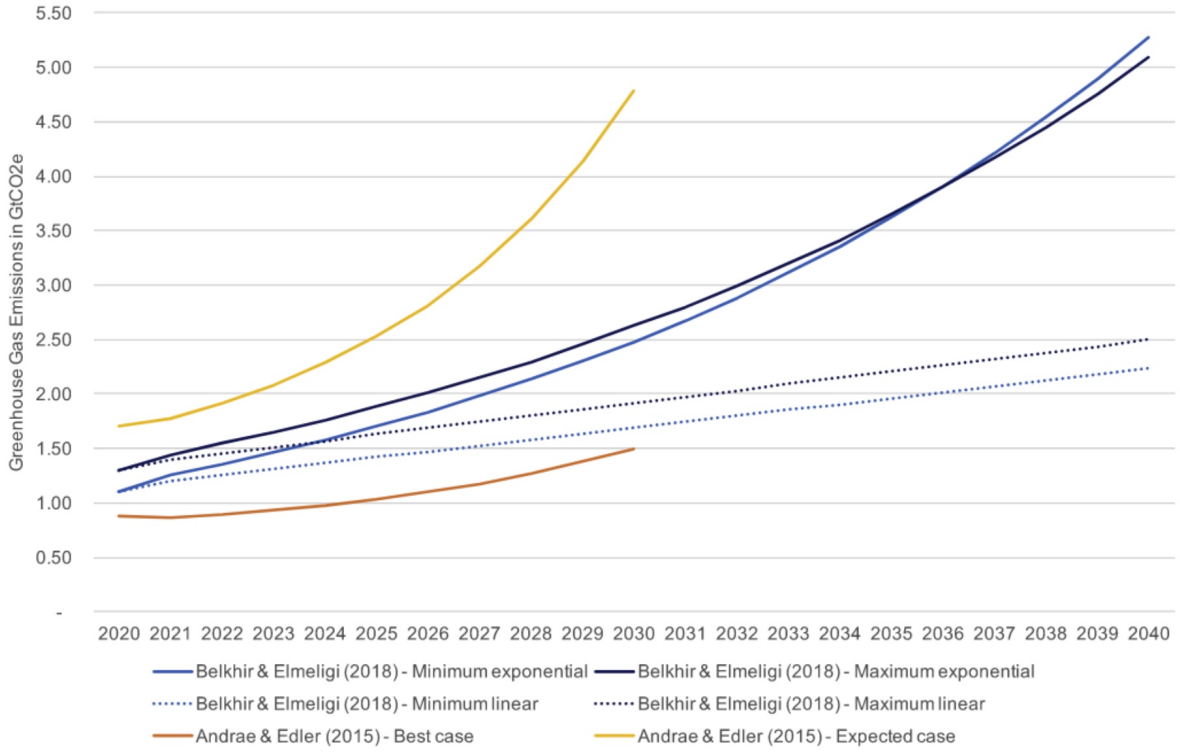



Figure 13: Projections of ICT's GHG emissions from 2020 studied by Freitag et al.

Contrastingly, Malmodin and Lundén anticipate a plateau in ICT emissions, driven by a saturation in market growth, exemplified by the smartphone market, where they argue that once a majority of the global population owns a smartphone, further growth—and thus emissions—will stabilize. However, Freitag et al. counter this by highlighting the industry's drive to avoid saturation and point to the rapid expansion of the Internet of Things (IoT) and Artificial Intelligence (AI) as key drivers of future emissions. The IoT sector alone is expected to see a dramatic increase in connected devices, growing from 15 billion in 2015 to an estimated 75 billion by 2025 [23], driving both embodied and operational emissions due to the vast infrastructure required to support these devices. Additionally, AI is projected to grow significantly, with the computational cost of AI training doubling every three months, which, coupled with the exponential increase in global data generation, could accelerate the environmental impact of ICT. Therefore, ongoing monitoring and evaluation of these emerging technologies are critical to understanding and mitigating their potential environmental impacts.

#### 4.0.3 AI's Impact on the Environment

The environmental impact of Artificial Intelligence (AI) remains largely underexplored, primarily due to the complex and multifaceted nature of AI systems. To date, no comprehensive environmental assessments have been conducted specifically for AI as a distinct domain. However, AI's impacts are generally considered within the broader context of Information and Communication Technology (ICT), as AI relies on similar hardware and has comparable environmental implications, particularly concerning greenhouse gas (GHG) emissions during its use phase. Nevertheless, as we have previously seen, many studies on ICT's emissions neglect, or vastly undermine, the impacts of AI. What sets AI apart within the ICT sector is its increasing reliance on vast amounts

of data and computational power, which contribute to its significant environmental footprint. The rapid growth of AI, especially in terms of compute-intensive processes, data consumption, and the associated energy use, has brought about serious environmental concerns.

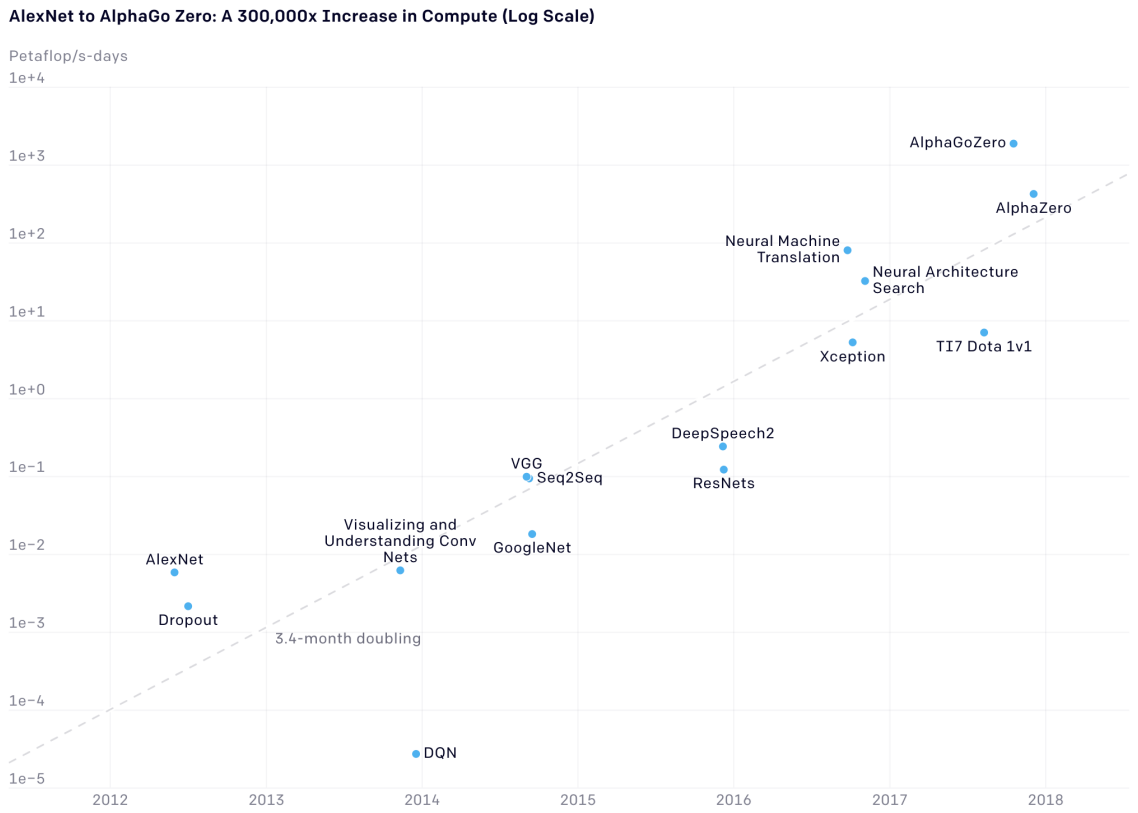



Figure 14: The total amount of compute, in petaflop/s-days used to train big relatively well-known AI models over the years, OpenAI [24]

For example, the computational demands for AI research have been doubling every few months, leading to a significant increase in energy consumption and carbon emissions. Figure 14 highlights this trend, showing a 300,000-fold increase in the computational power used to train AI models from 2012 to 2018. This growth is driven by three key factors: algorithmic innovation, the availability of data, and the amount of compute available for training. While algorithmic advancements and data are difficult to quantify, compute provides a measurable input to AI progress, often correlating with improved performance. The relevant metric here is not the speed of individual GPUs or the capacity of data centers but the total compute used to train a single model. As researchers have continued to leverage more parallelism and invest in custom hardware like GPUs and TPUs, the computational power required for training AI models has surged, pushing the boundaries of what is technologically and economically feasible. Although there are physical and economic limits to this trend, the current trajectory suggests that the demand for more compute-intensive AI models will continue to grow. Additionally, AI's dependency on specialized hardware such as GPUs and TPUs, which drive the expansion of data centers and contribute to embodied emissions, highlights the pressing need for a more detailed and dedicated assessment of AI's environmental impact.

Despite the growing awareness of these issues, environmental reporting in AI research is still in its infancy. Efforts to measure the carbon footprint of AI, particularly in the context of large-scale Natural Language Processing (NLP) models, have only recently gained traction through scientific papers such as [25]. The introduction of concepts like "Green AI" [26] and the development of tools for measuring energy consumption and GHG emissions are steps in the right direction, but they remain limited in scope. A study by Henderson et al. [27], which reviewed 100 papers from the 2019 NeurIPS conference, revealed that very few provided any form of environmental metrics, and none reported carbon emissions. This highlights the inadequacy of current reporting practices and underscores the need for more comprehensive assessments that go beyond just energy consumption to include factors like resource depletion, water usage, and biodiversity impacts.

To address these gaps, Ligozat et al. [28] advocate for the application of Life Cycle Assessment (LCA) methodologies to AI services. LCA is a well-established framework for evaluating the environmental impacts of products and services across their entire lifecycle, from production to end-of-life. Standardized by organizations like ETSI and ITU [29], and guided by ISO standards 14040 and 14044, LCA considers multiple environmental criteria, including but not limited to, climate change, ozone depletion, human toxicity, and resource depletion. The LCA methodology proposed by Ligozat et al. emphasizes the need for a comprehensive evaluation of AI services, which should include all stages of the AI lifecycle, such as data acquisition, model training, deployment, and disposal. The authors argue that the current focus on carbon footprint alone is insufficient and that other metrics, like water usage and abiotic depletion, must also be considered to fully understand AI's environmental impact.

A significant challenge in applying LCA to AI lies in the scarcity of publicly available data on the environmental impacts of AI hardware, particularly GPUs and TPUs, which are critical to AI operations. Ligozat et al. suggest that the AI community should advocate for greater transparency and push for companies to release more data on the environmental impacts of their hardware. This would enable AI practitioners to conduct more exhaustive and accurate environmental assessments of their models, thereby contributing to a more sustainable and environmentally-conscious AI industry.

#### 4.0.4 Predicting Energy Consumption & Carbon Footprint of AI

As Artificial Intelligence (AI) continues to grow in complexity and application, understanding and predicting its energy consumption has become increasingly important, especially given the environmental implications. Several researchers have proposed methods to estimate the energy requirements of AI models, each with its strengths and limitations. These methods are essential for assessing the environmental footprint of AI and for developing strategies to mitigate its impact.

One of the simpler approaches for predicting the energy consumption of AI models is the method of prediction by partial measurement. This technique, as discussed by Anthony et al. [30], involves using tools like Carbon Tracker, which predicts total energy consumption by averaging the energy used in previous epochs of the AI model's training. This method offers a straightforward way to estimate energy use based on historical data, making it useful for ongoing projects where past consumption patterns

are available. However, it may lack precision for new models or those with significant differences in architecture or scale from past iterations.

Another commonly used approach is the Thermal Design Power (TDP) approximation. This method, utilized by tools like ML CO2 Impact (Alexandre et al. [31] and Green Algorithms by Lannelongue et al. [32]), estimates the energy consumption of AI models by leveraging the TDP value, a manufacturer-provided metric that indicates the maximum amount of heat generated by a component under a steady workload. The total energy consumption can be approximated by multiplying the TDP by the time (t) the processor runs, as expressed in the equation $E_{total} = TDP \times t$. While this method provides a quick estimation, it has limitations. It assumes that the TDP accurately reflects power consumption during all phases of execution, which is not always the case, as some operations may be more computationally intensive while others may leave the processor nearly idle. Furthermore, without knowing the exact execution time beforehand, the predictive capability of this method is restricted.

Understanding the concept of Floating Point Operations Per Second (FLOPs) is crucial when discussing energy consumption predictions, particularly when linked to AI training. FLOPs represent the number of operations that a system can perform per second (being an operation an addition or multiplication, for example), serving as a measure of computational complexity. Schwartz et al. [26] advocate for using FLOPs as a more accurate metric than execution time to describe the workload of AI models. FLOPs are hardware-agnostic, meaning they can be used to assess the computational demands of a model regardless of the specific hardware it runs on. This metric is particularly useful when predicting the energy consumption of AI models during training, as it directly correlates with the intensity of computational tasks. Predicting training consumption with FLOPs is an approach explored by Mehta et al. [33], where the energy required to train a Deep Neural Network (DNN) is estimated based on the number of FLOPs necessary for a full forward and backward pass. The energy consumption for each pass is calculated by counting the operations and assigning an energy cost to them. This method allows for a detailed estimation of training energy consumption, taking into account the specific operations involved in training the model.

To obtain a comprehensive measure of the energy consumed during AI model training, the Power Usage Effectiveness (PUE) metric is often employed. PUE, as defined by Brady et al. [34], is the ratio of total energy consumed by a data center to the energy used specifically for computing tasks. The formula for PUE is given by:

$$PUE = \frac{TotalFacilityEnergy}{ITEquipmentEnergy}$$

A lower PUE value indicates more efficient energy use, with values closer to 1 representing an optimized data center. However, a good PUE does not necessarily mean that the data center is eco-friendly; it only indicates that the energy is being efficiently used for computing tasks. To calculate the total energy consumption for AI training, the PUE is multiplied by the sum of energy consumed by DRAM, CPUs, and GPUs involved in the process, as shown in the equation:

$$E_{training} = PUE \times (E_{DRAM} + E_{CPU} + E_{GPU})$$

Finally, once the total energy consumption during training is determined, it can be translated into the Greenhouse Gas (GHG) emissions using the Carbon Intensity (CI)

metric. CI measures the environmental impact of the electricity used, factoring in how polluting the energy source is—coal-based electricity, for example, has a much higher CI than wind energy. The relationship between energy consumption and GHG emissions is captured in the formula:

$$m_{GHG-training} = E_{training} \times CI$$

This approach allows researchers and practitioners to quantify the environmental impact of AI models in terms of their carbon footprint, offering a pathway to more sustainable AI practices. By understanding and applying these methods, we can better assess the energy demands and environmental costs associated with AI, ultimately guiding the development of more efficient and eco-friendly AI technologies.

#### 4.0.5 Impact of LLM's

This section is centered around the study of the environmental impact of BLOOM [35]. BLOOM is a 176-billion parameter language model developed as part of the BigScience workshop, an initiative that lasted from May 2021 to May 2022. The model was trained using 1.6 terabytes of data in 46 natural languages and 13 programming languages, utilizing the computational resources of the Jean Zay computer cluster in France. The training process spanned 118 days, accumulating a total of 1,082,990 GPU hours. The methodology for estimating the carbon footprint of BLOOM follows the Life Cycle Assessment (LCA) approach, focusing on different stages from the manufacturing of the equipment used for training to the deployment of the model via an API. The primary areas of study include embodied emissions, dynamic power consumption, idle power consumption, and emissions from deployment and inference.

The embodied emissions associated with the BLOOM model pertain to the materials and processes involved in producing the computing equipment necessary for its training. The servers used for training, similar to HPE's ProLiant DL345 Gen10 Plus, have a production footprint of approximately 2500 kg of CO2eq, excluding the GPUs. The Nvidia A100 GPUs used in the servers are estimated to have an embodied carbon footprint of around 150 kg of CO2eq per unit. Over the 1.08 million hours of training, the embodied emissions totaled approximately 7.57 tonnes of CO2eq for the servers and 3.64 tonnes for the GPUs, contributing 11.2 tonnes of CO2eq to BLOOM's carbon footprint. These figures do not account for the embodied emissions of the broader computing infrastructure, such as network switches and cooling equipment, due to a lack of available data.

In addition to embodied emissions, the dynamic power consumption during training represents a significant portion of BLOOM's carbon footprint. This is calculated based on the thermal design power (TDP) of the GPUs and the carbon intensity of the energy grid used. The BLOOM model's training required 1.08 million GPU hours on Nvidia A100 GPUs, consuming 433,195 kWh of electricity. Given the carbon intensity of the energy grid at 57 gCO2eq/kWh, the dynamic power consumption resulted in 24.69 tonnes of CO2eq emissions. Furthermore, the idle power consumption, which includes the energy used by the broader computing infrastructure when it is not actively engaged in training, added an additional 14.6 tonnes of CO2eq to the overall carbon footprint. This reflects the energy required to maintain the computing nodes, network, and cooling systems during periods of inactivity.

The deployment and inference phase of the BLOOM model also contributed to its carbon footprint. During a test deployment on Google Cloud Platform (GCP), BLOOM was run on an instance with 16 Nvidia A100 GPUs over 18 days, handling an average of 558 user requests per hour. The total energy consumption for this period was 914 kWh, with the GPUs accounting for 75.3% of this consumption. Despite the lower energy consumption per GPU compared to training, this phase highlighted the ongoing energy demands of maintaining the model in an active state, even when not processing requests. The total emissions for this period amounted to approximately 340 kg of CO2eq, considering the carbon intensity of the GCP region.

| Model name | Number of parameters | Datacenter PUE | Carbon intensity of grid used | Power consumption | $CO_2$eq emissions | $CO_2$eq emissions × PUE |
|---|---|---|---|---|---|---|
| GPT-3 | 175B | 1.1 | 429 $gCO_2eq/kWh$ | 1,287 MWh | *502 tonnes* | 552 tonnes |
| Gopher | 280B | 1.08 | 330 $gCO_2eq/kWh$ | *1,066 MWh* | *352 tonnes* | 380 tonnes |
| OPT | 175B | *1.09* | *231g$CO_2eq/kWh$* | *324 MWh* | 70 tonnes | *76.3 tonnes* |
| BLOOM | 176B | 1.2 | 57 $gCO_2eq/kWh$ | 433 MWh | 25 tonnes | 30 tonnes |

Figure 15: Comparison of carbon emissions between BLOOM and similar LLMs. Numbers in italics have been inferred based on data provided in the papers describing the models [35].

Finally, Figure 15 compares the carbon emissions of BLOOM to other similar large language models (LLMs), such as GPT-3, Gopher, and OPT. BLOOM's total emissions were significantly lower, primarily due to the lower carbon intensity of the energy grid used during training. While BLOOM consumed slightly more energy than OPT, the overall emissions were much lower due to the more efficient energy grid. This table highlights the importance of considering both energy consumption and carbon intensity when comparing the environmental impact of different models.

# 5 Economic Impact

### 5.0.1 Impact on Productivity

Artificial Intelligence (AI) is increasingly recognized as a powerful tool for enhancing productivity across various sectors, driving both organizational efficiency and broader economic growth. The strategic management of AI innovation, which includes the adoption of AI tools, project management, and the integration of AI into business processes, plays a critical role in realizing these productivity gains. By automating routine tasks, improving decision-making processes, and fostering innovation, AI innovation management not only increases efficiency but also enhances competitiveness. However, the successful implementation of AI requires addressing several challenges, including the need for employee retraining, ensuring data privacy, and navigating ethical considerations [36]. This comprehensive approach to managing AI can significantly boost organizational productivity, thereby contributing to economic growth, particularly in emerging economies that are poised to benefit from AI-driven advancements [37](Partridge, Tsvetkova, & Betz, 2020). Much of the information in this section is drawn from the study "The Impact of AI Innovation Management on Organizational Productivity and Economic Growth: An Analytical Study" by Zhaoxia Yi and Shirley Ayangbah [38].

The impact of AI on productivity is evident across a range of industries, with numerous case studies highlighting the specific mechanisms through which AI enhances operational efficiency and reduces costs. In the manufacturing sector, AI technologies are integrated

into various processes to optimize production and improve efficiency. For instance, predictive maintenance powered by AI algorithms can anticipate equipment failures before they occur, reducing unexpected downtimes and maintenance costs. Additionally, AI-driven quality control systems detect product defects with high precision, ensuring greater consistency and higher product quality. AI's role in supply chain optimization is also significant, helping companies manage inventories, forecast demand, and streamline logistics, thereby reducing overhead costs and improving operational efficiency [39] (Kim et al., 2021).

In the finance sector, AI has revolutionized decision-making processes, risk management, and customer service. AI-driven analytics enable financial institutions to process large volumes of data rapidly and accurately, improving investment decisions, fraud detection, and customer relationship management. For example, Mastercard's AI-based fraud detection system analyzes transaction data in real-time to identify potentially fraudulent activities, significantly reducing false positives and enhancing the accuracy of fraud detection, resulting in substantial cost savings [40](Carneiro et al., 2018). Similarly, BlackRock has integrated AI into its investment decision-making process, where AI systems analyze market data, financial reports, and economic indicators to optimize portfolio performance, leading to higher returns for clients. These examples underscore the transformative impact of AI in finance, where automated processes reduce the need for manual intervention, speeding up transactions and reducing errors, ultimately leading to enhanced productivity [41] (Criscuolo et al., 2020).

The healthcare sector also benefits significantly from AI integration, particularly in diagnostics, treatment planning, and patient care management. AI applications in medical imaging analysis, personalized medicine, and predictive analytics improve the accuracy and efficiency of healthcare services. For instance, IBM Watson Health has been used by the University of North Carolina School of Medicine to assist in cancer diagnosis and treatment planning, providing personalized treatment recommendations that lead to improved patient outcomes and reduced costs [42](Somashekhar et al., 2018). Moreover, AI-powered drug discovery, as seen in Pfizer's collaboration with Insilico Medicine, accelerates the identification of potential drug targets by analyzing vast amounts of genomic and clinical data, reducing the time and cost associated with traditional drug discovery methods [43](Zhavoronkov et al., 2019). These case studies demonstrate the potential of AI to enhance productivity across diverse sectors, contributing to better outcomes and operational efficiencies.

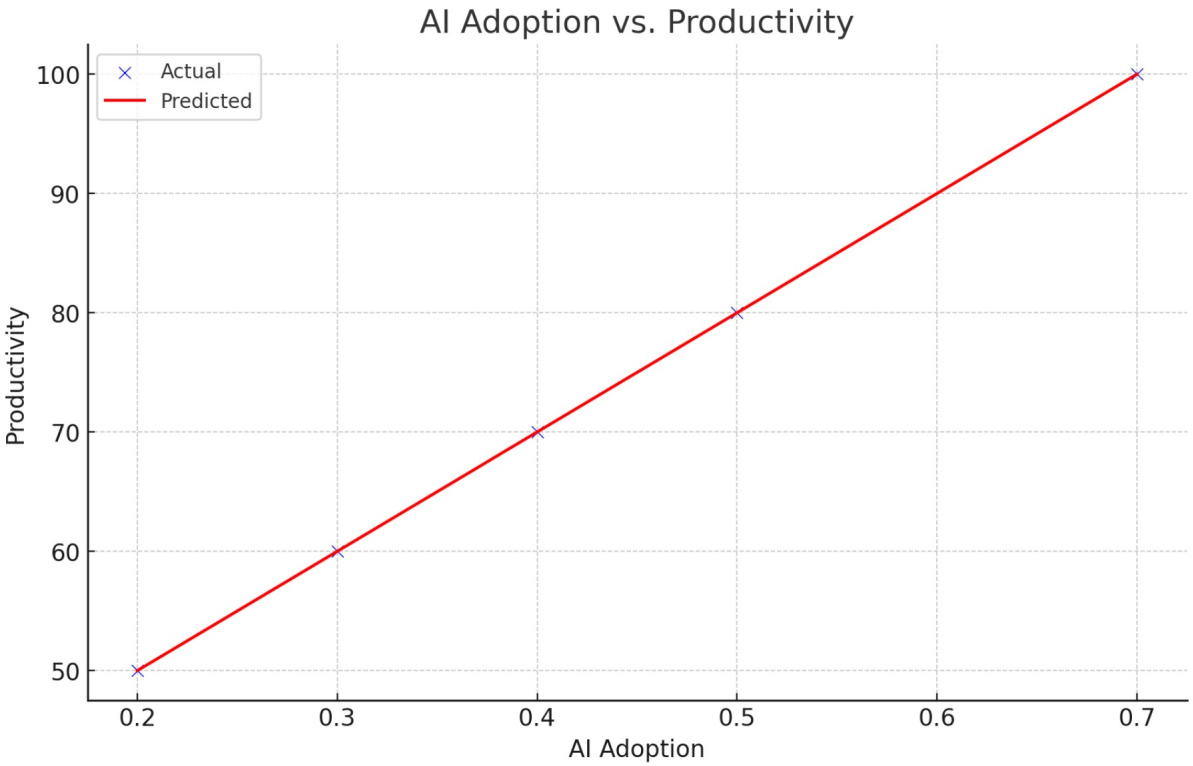



Figure 16: AI adoption vs. Productivity [44]

Overall, AI's impact on productivity is profound, offering opportunities for significant improvements in efficiency, cost reduction, and service quality across various industries. The relationship between AI adoption and productivity is notably positive, as depicted in Figure 16. As organizations increasingly adopt AI technologies, they experience improvements in productivity due to enhancements in product development cycles, more effective risk management, and the creation of new AI-driven features. High-performing organizations that deeply integrate AI into their core business functions tend to see substantial productivity gains. However, the widespread adoption of AI also necessitates careful consideration of the challenges associated with its implementation, such as the need for workforce retraining, the importance of maintaining data privacy, and addressing ethical concerns. By effectively tackling these challenges, organizations can fully capitalize on the benefits of AI, leading to sustained economic growth and enhanced competitiveness in a rapidly evolving technological landscape [45] (Brynjolfsson and McAfee, 2020).

#### 5.0.2 Broader Economic Implications

Artificial Intelligence (AI) has become a crucial catalyst for economic growth, significantly impacting productivity and efficiency across various sectors. As organizations increasingly integrate AI technologies, they are able to enhance their production capabilities, offering more goods and services at reduced costs, which in turn boosts economic output. This trend is particularly evident in emerging economies, where AI-driven advancements have helped counteract productivity slowdowns in more developed nations, thereby contributing to global GDP growth [46](Esfahani et al., 2020). These productivity improvements through AI are instrumental in driving economic progress by optimizing operations, reducing operational expenses, and enabling businesses to meet market demands more effectively. However, alongside these benefits, challenges such as workforce

retraining, data privacy, and ethical considerations must be addressed to ensure that the advantages of AI are shared equitably.

A significant concern related to AI's role in economic growth is the potential exacerbation of income inequality. The distribution of AI benefits is not always uniform, which can widen the gap between those with access to AI technologies and those without. Research suggests that while AI contributes to productivity and economic growth, it can also increase disparities in income and wealth [47](Seo et al., 2020). This growing inequality can adversely affect investment and productivity, potentially hindering broader economic development. It is therefore essential to implement strategies that promote equitable access to AI technologies and mitigate the risks associated with income inequality. These strategies might include education and training programs to help workers adapt to new technologies and ensure that AI-driven advancements are accessible across all sectors of society.

In terms of policy measures and corporate strategies, it is crucial to focus on tailored policies that enhance workforce adaptability and promote widespread access to AI technologies [48] (Hailemariam and Dzhumashev, 2019). Effective management of AI innovations not only drives productivity but also supports long-term economic development by fostering technological progress, a key driver of endogenous economic growth [49] (Romer, 1990). Policymakers and industry leaders must create an environment that encourages AI innovation while also addressing the ethical and legal challenges that accompany its use. This balanced approach can maximize AI's positive impact on productivity and economic growth while minimizing potential risks.

The broader implications of AI on the economy extend beyond individual businesses, affecting entire industries and national economies. In manufacturing, AI has optimized production processes, lowered costs, and increased efficiency, resulting in greater economic output and enhanced industrial competitiveness. In the finance sector, AI has revolutionized decision-making, risk management, and customer service, contributing to economic stability and growth. In healthcare, AI has improved patient outcomes, reduced costs, and increased the efficiency of medical services. These examples highlight AI's transformative potential across various sectors and its capacity to drive economic growth by enhancing productivity and operational efficiency.

To fully realize the economic benefits of AI, it is essential to draw on lessons learned and best practices from different industries. In manufacturing, key areas of focus include workforce retraining and the integration of AI with existing infrastructure. In finance, the emphasis should be on implementing strong data security measures and ensuring regulatory compliance. In healthcare, specialized training for professionals and addressing data privacy issues are critical for the successful integration of AI. By adopting these best practices, organizations can ensure that AI technologies are implemented effectively, leading to sustained economic growth and improved competitiveness in an increasingly technological world.

#### 5.0.3 Impact of AI on the Labour Market

The following section sums up the study developed by Michael Webb on his paper: "The Impact of Artificial Intelligence on the Labour Market" [50]. Artificial Intelligence (AI) is increasingly recognized as a transformative force in the labor market, reshaping

the demand for various occupations while simultaneously enhancing productivity and living standards. As AI technologies, particularly machine learning algorithms, achieve superhuman performance across a range of tasks—spanning from high-wage jobs like radiologists to low-wage positions such as agricultural workers—their impact on job displacement and income inequality has become a pressing concern. To better understand these distributional effects, a novel methodology was developed by Michael Webb [50] identify which tasks within occupations are most susceptible to automation by AI. This method quantifies the "exposure" of various jobs to AI by analyzing the overlap between the tasks described in job descriptions and the tasks performed by AI as documented in patents. This approach not only provides insights into which occupations are most vulnerable to AI-driven automation but also allows for the estimation of the potential impacts of AI on wage inequality, building on historical data from previous technological shifts involving software and robots. The results indicate that AI exposure is particularly high in high-skilled occupations, suggesting that the effects of AI on the labor market will differ significantly from those of earlier technologies. This method offers a comprehensive and objective way to measure the potential impacts of AI across the economy, aiding policymakers and businesses in anticipating and managing the changes brought about by AI.

In the first part of the paper the author proposes a mathematical model whose objective is to understand and identify the different channels through which technology (specifically AI) can affect labour demand. The model is based upon previous works from Acemoglu and Restrepo (2018) [51]. This model is a simple static model in which the objective is to understand how the technological advancement affects different parts of the labour market, specifically how it shifts the demand for specific occupations. The economy is modeled with a single firm that produces a distinct final product. This firm generates the product, $X$, by integrating the outputs of various occupations, $O_i$, using a production function characterized by a constant elasticity of substitution, denoted by the elasticity parameter $\rho$.

$$X = (\sum_i \alpha_i O_i^{\rho})^{\frac{1}{\rho}}$$

Each occupation, $O_i$, generates its output by combining the outputs from a variety of tasks. These tasks are represented as $T_{i,j}$, where $j$ indexes the specific tasks. The outputs of these tasks are integrated using another constant elasticity of substitution production function, which has a different elasticity parameter, $\rho_t$ (with $t$ indicating "task"):

$$O_i = \left( \sum_j \alpha_j T_{i,j}^{\rho_t} \right)^{\frac{1}{\rho_t}}.$$

Certain tasks are capable of being automated, while others are not. Tasks that can be automated might be executed either by humans, $H$, or by machines, $R$. At the task level, humans and machines serve as perfect substitutes. Tasks that cannot be automated must be performed solely by humans:

$$T_{i,j} = \begin{cases} H_{i,j} + A_{i,j} R_{i,j} & \text{if automation is feasible;} \\ H_{i,j} & \text{otherwise.} \end{cases}$$

Human productivity is normalized to one, but machine productivity can improve over time (for a specific feasible task). Machine productivity is denoted by $A_{i,j}$.

The firm considers the wage of human labor in occupation $O_i$, denoted as $w_i$, as a given. Similarly, the rental rate of a machine is given by $r$. In this setup, the firm will exclusively employ human labor for task $j$ in occupation $i$ if $w_i < \frac{r}{A_{i,j}}$, and will use machines exclusively if the inequality is reversed. If automating a task is not feasible, this corresponds to $A_{i,j} = 0$.

By running several simulations to identify the impact of machine productivity in human labour demand, two channels can be identified:

- Let there be an economy in which the product X needs of a single occupation composed of two tasks, one can be automated and the other one not. There are two levels of demand to consider, the firm's demand of labour per unit of product, and the total consumer's demand of said product. If machine productivity increases, the firm's demand of human labour will decrease for the task that can be automated. Moreover, if the tasks are substitutes, the demand will decrease even more. Nonetheless, this productivity increase will cause the cost of production (labour) to decrease and therefore the price of the product to decrease too, which can lead to an increase of demand. To meet the increasing demand production is increased and thus the demand of labour, to the point that this opposing effects can result in a net zero or even a net increase in human labour demand.

- Let there be another case, where there are two occupations each one composed of two tasks. Only the first task of the first occupation can be automated. As depicted in Figure 17, the demand of human labour in occupation 1 is ambiguous and depends on the elasticity value of the occupations. Strong elasticity values $\rho = 0.8$ where the occupations can be substituted, lead to an increase on human labour. Negative elasticity values $\rho = -10$ where the occupations are complementary, lead to a decrease of human labour with machine productivity. Notice that in this case the elasticity parameter between tasks is $\rho_t = -10$ so tasks are complementary. When the two occupations are substitutes, as machine productivity increases ($A_{1,1}$), occupation 1 becomes less expensive and therefore its demand increases over occupation 2. This causes the human labour necessary to complete task 2 (from occupation 1) to increase while the demand of this kind of labour decreases for occupation 2, and task 1 of occupation 1. Once again, this contradictory forces can cause a net increase of human labour demand, for $\rho > 0.5$.

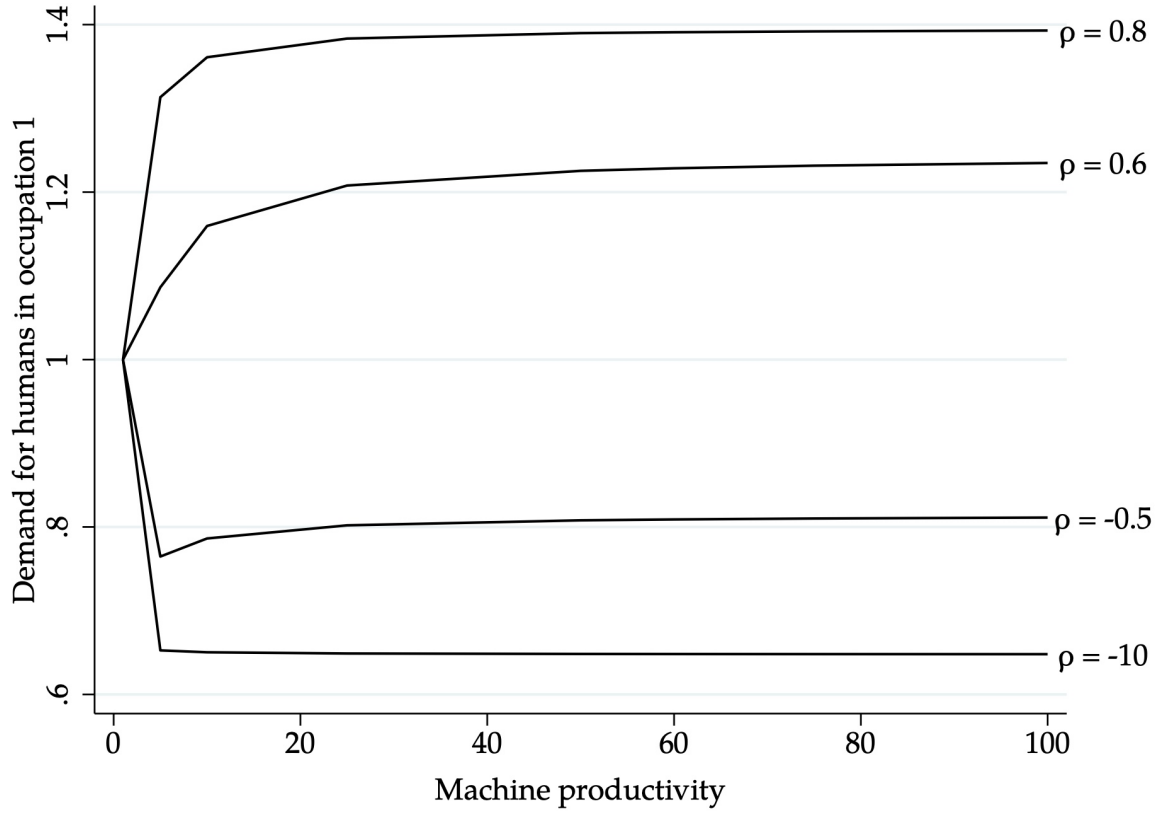



Figure 17: Model simulation results: demand for humans in occupation 1 (occupation being auto- mated) as a function of robot productivity, for different values of firm-level elasticity of substitution parameter . Task-level elasticity of substitution parameter t is fixed at 10 [50]

The analysis demonstrates the intricate relationship between displacement and productivity effects within an economy characterized by a single firm. The productivity impact is observed on two distinct levels: within the firm itself and in the broader market. Internally, the substitutability between different occupations in the production process influences how automation alters the firm's demand for specific occupations per unit of output. Externally, changes in the price of the final product affect consumer demand, further complicating the overall effect of task-level automation on occupational demand. This complexity results in a theoretically ambiguous impact on labor demand, making it difficult to predict the direction of impacts without strong assumptions about the elasticity of substitution parameters across various sectors. The method developed allows for the identification of tasks where technology can effectively replace human labor, facilitating the empirical estimation of the relationship between the extent of task automation and subsequent changes in demand for specific occupations. Given the theoretical uncertainty inherent in the model, this relationship is critical for understanding the potential impacts of AI. Historical case studies (robots and software) indicate a negative relationship between the extent of task automation and changes in employment and wages across occupations, suggesting that, until now, the forces favoring negative correlation have prevailed. This evidence suggests that similar outcomes may be expected with AI in the future.

To evaluate the exposure of occupations to specific technologies, the approach involves analyzing patent texts to identify the tasks that a given technology can perform and

then measuring the extent to which each occupation in the economy involves performing similar tasks. This process begins by selecting a set of patents corresponding to a particular technology, such as artificial intelligence, identified through specific keywords in their titles or abstracts. From these patent titles, verb-noun pairs are extracted using a dependency parsing algorithm (Honnibal and Johnson 2015) [52], which then are lemmatized to ensure consistency. Similarly, for occupations, tasks are described in natural language and verb-noun pairs are extracted from these descriptions. The exposure score for each occupation is calculated by matching these pairs with those found in the patents, with adjustments made using WordNet (Miller 1995) [53]to group nouns into conceptual categories to account for variations in generality. This method allows for a systematic comparison between the tasks performed by humans in various occupations and those described in technology patents, providing a measure of the potential for technology to substitute for human labor.

The method to calculate an occupation's exposure score to a given technology relies on analyzing the frequency of specific verb-noun pairs extracted from task descriptions and comparing them with similar pairs derived from patent titles. For a given technology $t \in T$, let $f_c^t$ represent the raw count of occurrences of the aggregated verb-noun pair $c$ in the patent titles for technology $t$. The total set of such pairs is denoted by $C^t$. The relative frequency, $r_c^t$, of each pair $c$ in the patent titles for technology $t$ is calculated as:

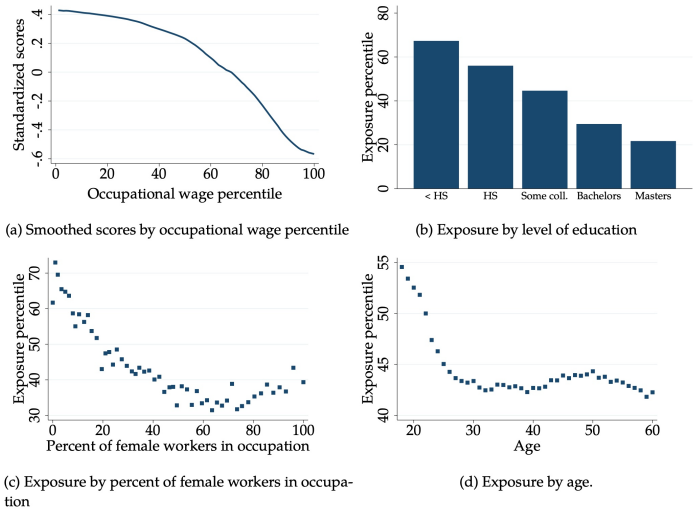
$$r_c^t = \frac{f_c^t}{\sum_{c \in C^t} f_c^t}$$

These task-level scores for AI-related patents are used to determine the exposure of each occupation. The overall exposure score for an occupation $i$ to a technology $t$ is then computed by taking a weighted average of these task-level scores:

$$\text{Exposure}_{i,t} = \frac{\sum_{k \in K_i} \left[ w_{k,i} \cdot \sum_{c \in S_k} r_c^t \right]}{\sum_{k \in K_i} \left[ w_{k,i} \cdot |c : c \in S_k| \right]}$$

In this equation, $K_i$ denotes the set of tasks within occupation $i$, and $S_k$ refers to the set of verb-noun pairs extracted from task $k \in K_i$. The weight $w_{k,i}$ represents an average of the frequency, importance, and relevance of task $k$ to occupation $i$, as specified in the O*NET database, with weights normalized to sum to one. This exposure score effectively measures the intensity of patenting activity related to the tasks within each occupation, thereby indicating the potential impact of a given technology on those tasks.

The paper [50] continues by applying the methodology to two historic technological revolutions: robots and software. In both cases the exposure coefficients are computed to understand which occupations are expected to suffer a greater impact by these technologies. Then the demographic characteristics of the workers enrolled in the most exposed occupations are studied. Figures 18 and 19 show the results obtained.



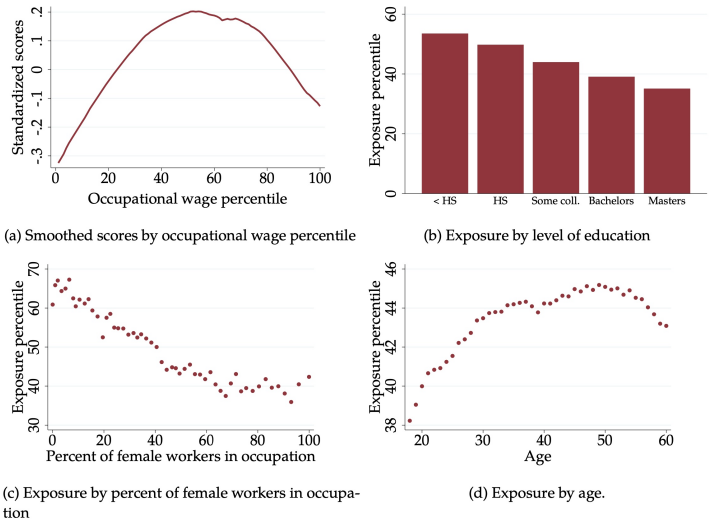

(a) Smoothed scores by occupational wage percentile

(b) Exposure by level of education

(c) Exposure by percent of female workers in occupation

(d) Exposure by age.

Figure 18: Model simulation results: Exposure to robots by demographic group [50]




(a) Smoothed scores by occupational wage percentile

(b) Exposure by level of education

(c) Exposure by percent of female workers in occupation

(d) Exposure by age.

Figure 19: Model simulation results: Exposure to software by demographic group [50]

As the Figures show, the robot and software technological revolutions distinctly shaped the labor market, with varying impacts on different demographic groups. Robots primarily disrupted lower-wage occupations, particularly those involving physical and repetitive tasks—jobs often held by younger males with lower educational attainment. This shift underscores the susceptibility of manual labor to automation, where tasks can be mechanized efficiently. In contrast, the software revolution had a more pronounced effect on middle-wage occupations that required technical skills and a higher level of education. These jobs, often in administrative or operational roles, were increasingly automated as software systems took over data management, processing, and routine decision-making tasks. Interestingly, both technological waves had limited impact on highly complex and well-paid occupations. These roles, typically held by individuals with advanced education and specialized skills, involve cognitive tasks that are less routine and more analytical or creative, making them harder to automate. This resilience in high-skill jobs highlights the continuing importance of human expertise in areas that require judgment, creativity, and complex problem-solving—skills that current

technologies, whether robots or software, cannot easily replicate. Moreover, the demographic analysis reveals an important aspect of technological change: while automation can displace workers, it also creates a demand for new skills and new jobs. The challenge, therefore, lies in managing this transition, ensuring that workers in at-risk occupations are retrained and equipped to move into roles that are less susceptible to automation. This shift not only affects wage distribution but also has broader implications for social equity and economic stability. The same study was then performed with the AI revolution, and the results are presented in Figure 20.

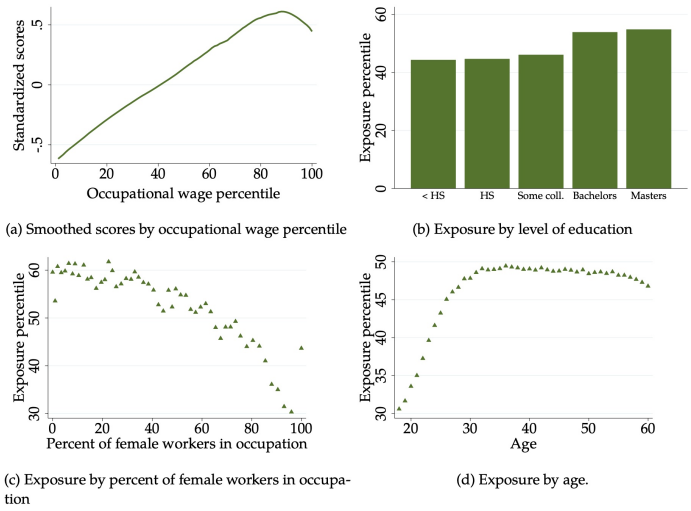



Figure 20: Model simulation results: Exposure to AI by demographic group [50]

The advent of advanced AI technologies, such as GPT-4, marks a significant shift in the technological landscape, particularly due to their capabilities in performing complex cognitive tasks. Unlike previous technological revolutions, which primarily impacted lower and middle-wage occupations, AI is expected to influence higher-paid, education-intensive sectors. As Figure 20 indicates, these advanced models are expected to have a substantial impact on more senior positions, given their ability to extract intricate insights from vast datasets, engage in creative brainstorming, and solve multifaceted problems—capabilities traditionally associated with specialized professions. This evolution suggests that AI will have a more profound impact on roles requiring significant cognitive input, including those in research, consulting, legal analysis, and executive decision-making. As AI tools become more integrated into these professions, they may complement human workers by handling routine data analysis and generating preliminary insights, allowing professionals to focus on higher-level strategic tasks. However, there is also the potential for displacement, as AI could increasingly take over tasks that were previously thought to be the exclusive domain of highly educated and experienced individuals. Furthermore, this trend might lead to a reconfiguration of job roles, where professionals in high-wage sectors might need to adapt by acquiring new skills that emphasize the human aspects of their work—such as emotional intelligence, ethical decision-making, and creative thinking—that AI cannot easily replicate. The challenge for these industries will be to harness AI's capabilities while ensuring that human expertise remains central, thus maintaining the value and relevance of high-skill jobs in an AI-driven economy.

The paper also shows that exposure to robot substitution has a substantial negative correlation with both wages and employment within the most affected occupations.

Analyzing the data from the manufacturing sector over a thirty-year period, it was found that occupations with higher exposure to robot technology—measured by moving from the 25th to the 75th percentile in exposure—experienced significant declines in industry employment shares, ranging between 9% and 18%. Similarly, wages in these occupations fell by approximately 8% to 14%. These findings highlight that the decline is not merely due to broader trends in manufacturing but is concentrated within specific occupations that have been directly impacted by automation. Furthermore, when controlling for various factors such as offshoring, changes in consumer demand, and shifts in workforce education levels, the negative impacts of robot exposure remain robust, though their magnitude may be somewhat reduced. This suggests that the adoption of robots has had a unique and profound effect on labor demand within these industries. Additionally, the software revolution, which similarly automated many middle-skill jobs, shows comparable results, with exposed occupations witnessing a decline in both employment and wage growth. These outcomes underline the disruptive potential of automation technologies and the importance of understanding their specific impacts on different sectors of the labor market.

Finally, the author tries to predict the potential impacts of AI on wage inequality by extrapolating from the previously discussed trends observed with software and robots. Assuming that AI exposure has a similar negative and approximately linear relationship with wage changes, the study estimates how wage distributions could shift. By adjusting the average wages of occupations based on their AI exposure scores and a range of coefficients derived from historical data, the results suggest that AI could reduce wage inequality in the middle of the distribution (as indicated by a decline in the 90:10 wage ratio) but may increase inequality at the top (as shown by a rise in the 99:90 ratio). Specifically, if AI follows the same exposure coefficient as software, a 4% decrease in 90:10 inequality is expected, while with the coefficient from robots, the decrease could be as much as 9%. This suggests that while AI may compress wages for mid-level occupations, it could exacerbate disparities at the highest levels, where occupations are less exposed to AI.

# 6 Development of the algorithms and results

As explained in the Objectives and Problems Addressed by the Project, this section aims to develop an algorithm based upon Convolutional Neural Networks capable of segmenting camouflaged animals in constrained environments. This process will show the complexity associated with developing modern AI models capable to perform complex tasks, showing the different steps necessary to develop such technologies. The resulting algorithm and its performance will also be discussed, as well as the costs necessary to develop such technology and a rough estimation of the GHG emissions produced on its development and training.

## 6.1 Data pre-processing

When developing a dataset of images to train a CNN, the first step is formatting the animal images into a multi-dimensional numpy array so that they can be fed into the neural networks. Given that this kind of networks use supervised learning, each image used for training must be paired with another image in which only the desired output of the model is present (mask). Therefore both the colored images and the binary masks must be resized, and several functions must be coded in order to create the arrays.

Specifically:

- Each colored image is resized to a 256x256x3 array where the last 3 dimensions correspond to the RGB coding system, where each pixel has 3 values between 0-255 corresponding to the red, green and blue channels. Then images from the same dataset are integrated into a single array.
- Masks are resized following a similar logic, but with a binary encoding where the value 0 corresponds to a black pixel (background) and 1 to a white one (animal). Therefore each mask has the following shape 256x256x1. All masks from the same dataset are integrated in the same array.
- Thus for a single dataset there are 2 arrays, one for the images and one for the masks. It very important to notice that the image number n of the first array corresponds to the mask number n of the second array. Also, the arrays must be shuffled, keeping this index correlation, in order to reduce biases during training.

#### 6.1.1 Camouflaged Dataset ($\mathbf{D}_{CAM}$)

The dataset provided by CS Group, CAMO [14] was composed of several types of images:

- Colored images
- Images with the contour of the animals
- The animal's shadows (masks/ground-truth)
- Animals shadows but with several instances (for the cases where more than an individual animal is present in the image)

There were 10 000 images from each category, where 5 067 images corresponded to camouflaged images and the rest were non-camouflaged images. Only the camouflaged images were kept since the masks were completely black images for the non-camouflaged ones. Some images were also corrupted so they were neglected. As a result, only 3758 camouflaged images were kept on the CAM dataset. 80% of them were used for training, and the other 20% for validation.

#### 6.1.2 Non Camouflaged Dataset ($\mathbf{D}_{NONCAM}$)

Non-camouflaged images were selected as a first step toward the final objective of segmenting camouflaged images. This approach was taken to develop robust algorithms and to ensure that any errors encountered during the process were not due to the inherent complexity of camouflaged images. By starting with simpler visuals, it was possible to accurately assess and improve algorithm performance before tackling the more challenging task of segmenting camouflaged images.

The Oxford-IIIT Pet dataset [54] was chosen since it is a relatively big dataset composed of simpler images, that include only dogs and cats. It is a very used and classic dataset for segmentation tasks. Nevertheless the masks that it integrates are trimaps where the contours are in white, the shadow of the animal in black and the background in grey. This format is not adapted to the models developed in this project, since they were coded to take as input binary masks. Therefore, the images were transformed into the correct format.

The dataset is composed of 7 389 colored images with their respective masks. Nevertheless, some of the images were corrupted so only 7383 images were used and the same amount of masks.

### 6.1.3 Mixed Dataset ($\mathbf{D}_{BOTH}$)

Given the scarcity of camouflaged dogs and cats present in the CAM dataset, the creation of a new dataset that incorporated both camouflaged and non-camouflaged images of dogs and cats was necessary. To increase the number of camouflaged images the dataset was artificially augmented by applying rotations to the images (resp. masks) as well as mirror effects. 301 images (resp. masks) from $D_{CAM}$ were extracted, 700 images (resp. masks) from $D_{NONCAM}$ dataset and finally 200 images from the augmented set were included, giving a total of 1201 images (resp. masks). This new dataset was called $D_{BOTH}$.

Figure 21: An image of cat from the $D_{BOTH}$ dataset: normal image, mirrored image and rotated image, from left to right, with the corresponding masks

## 6.2 Classification model

The Classification Model has been implemented by application of a **fine-tuning** of the **ResNet50** Model developed by Kaiming He et al. on the paper "Deep Residual Learning for Image Recognition" on 2015 [13]. ResNet50 is a convolutional neural network, that introduced the concepts of Residual Networks and Skip Connections.

Residual Convolution Blocks, present in Figure 25c) were introduced into ResNet to tackle the vanishing gradients problem . This problem , previously explained on section 2.3.1, occurs during the training of deep neural networks, where the gradients of the loss function with respect to the weights become extremely small. This causes the updates to the weights to be very small, effectively preventing the network from learning. As a result, the earlier layers in very deep networks learn very slowly or not at all, which hampers the overall training process and model performance. Residual blocks, introduce skip connections (or shortcuts) that allow the input to a layer to be directly passed to a subsequent layer, bypassing one or more intermediate layers. Therefore they ensure

that gradients can flow directly through the network, allowing them to remain large enough for effective back-propagation.

Particularly, ResNet50 has around 23.6 million parameters and has been trained on an extremely large dataset, imagenet-1k, that contains more than 1.2 million training images. Thanks to this huge training dataset, ResNet presents a high generalization capacity. This is the main reason for which this model (that is still a CNN) was chosen as the fine-tuning base for the classification algorithm.

Three layers have been added to the pre-trained model:

1. a flatten layer
2. a dense layer with 512 neurons and a ReLU activation function
3. a dense layer with 5 output neurons, each corresponding to a possible class for the camouflaged images: **Amphibian, Aquatic, Flying, Terrestrial and Other**

Only these last three layers have been trained through back-propagation using the Camouflaged Dataset (CAM Dataset), since the rest of the model has been frozen in order to preserve its feature detection capabilities. The training has been done using Adam as a compiler and Categorical Cross Entropy as the loss function, which is explained later on. The model has been trained on 10 epochs but stopped at the sixth one because of an early stopping callback that aimed to prevent it from over-fitting. The early stopping should cause the training to stop if the validation loss has not decreased in the last 5 epochs (which corresponds to the patience). It is important to notice that only the model at its most performing state will be kept.

Categorical cross-entropy is a loss function that measures the performance of a classification model whose output is a probability value between 0 and 1. It compares the predicted probability distribution over classes with the actual distribution, which is typically represented as a one-hot encoded vector (where the correct class is 1 and all others are 0). It can be expressed as:

$$L = -\sum_{i=1}^{N} y_i \log(p_i)$$

where

- $N$ is the number of classes.
- $y_i$ is a binary indicator (0 or 1) if class $i$ is the correct classification.
- $p_i$ is the predicted probability for class $i$.

The goal of this loss function is to minimize the distance between the predicted probability distribution and the true distribution. It penalizes the model more when the predicted probability for the correct class is low.

## 6.3 Segmentation models

### 6.3.1 U-Net

U-Net is a Convolutional Neural Network published in 2015 in [55] that aims to perform highly precise segmentations of Biomedical images with relatively small training datasets. This precise characteristic of the model is the key feature for which the U-Net model was implemented in this project. The scarce amount of images in the datasets employed in this study (compared to the necessary images to train other state-of-the-art models) and the difficulty of segmenting camouflaged images favoured the choice of U-Net.

The network comprises a contracting path and an expansive path, forming its characteristic U-shape. The contracting path functions as a typical convolutional network with repeated applications of convolutions, each followed by a rectified linear unit (ReLU) and a max pooling operation. During contraction, spatial information is reduced while feature information is increased. The expansive pathway then combines feature and spatial information through a sequence of up-convolutions and concatenations with high-resolution features from the contracting path. An overview of the model is present on Figure 22.

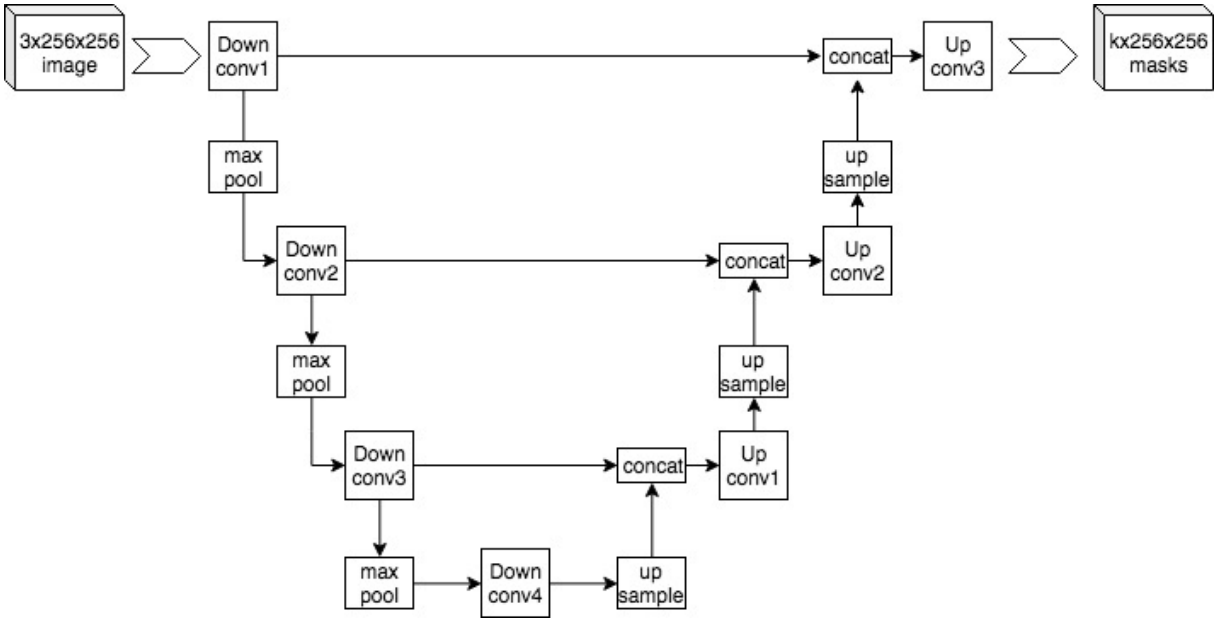



Figure 22: U-Net Model (in our case k=1)

U-Net incorporates several skip connections that cross from the contracting path to the expansion path. This skip connections allow the spacial information of the image (strongly present in the upper layers of the contracting path) to be incorporated into the deeper parts of the network, present into the expansion path (which are much richer in feature information). This structure results in high performing segmentations that are able to generalize with relative low training examples.

The specific implementation of U-Net on this project takes as an input an image 256x256 encoded in RGB format (256x256x3) and gives as output a binary image where the animal has been segmented (256x256x1). The model has 7 771 873 parameters. The contracting path incorporates 4 series of 2 convolutional blocks + max pooling in order to reduce the size of the images and extract the feature information. After each series the amount of feature maps present at each convolutional block doubles, starting at 32 and finishing at 256. At the bottom of the model there is another series known as the

bottleneck, in which there are 512 feature maps per convolution. The contracting path incorporates 4 series as well, where each series incorporates a transpose convolution (up sample) + concatenation (skip connection) + 2 convolution blocks that aim to increase the size of the image, incorporate the spacial information and keep extracting the feature information from the image. After each series the feature maps of each convolution reduce by half, starting at 256 and ending at 32. At last we apply a final convolution (1 feature map) + sigmoid function with the objective of the output being a binary image where each pixel has a value $\in [0, 1]$ representing the probability of that pixel being a part of the animal. Note that After each convolution a Batch Normalization is implemented to reduce the vanishing gradients problem as well as a Dropout with 0.1 rate to reduce over-fitting and to increment the generalization capabilities of the network.

The training has been done using Adam as a compiler and Binary Focal Loss as the loss function. The model has been trained on 20 epochs with an early stopping callback with patience 4 and a batch size of 16. Both Attention U-Net and R2 Attention U-Net have been trained with the same specifications.

For image segmentation, binary focal loss can be expressed as:

$$FL(y_i, \hat{y}_i) = -\alpha y_i (1 - \hat{y}_i)^\gamma \log(\hat{y}_i) - (1 - \alpha)(1 - y_i)\hat{y}_i^\gamma \log(1 - \hat{y}_i)$$

where $y_i$ is the true label and $\hat{y}_i$ is the predicted probability for pixel $i$. Binary focal loss helps the model focus on hard-to-segment pixels by reducing the loss contribution from well-segmented pixels. $\alpha$ balances the importance between foreground and background pixels. $\gamma$ focuses learning on difficult pixels by modulating the loss. This makes focal loss particularly effective for image segmentation tasks with imbalanced datasets, where foreground objects may occupy a small portion of the image compared to the background.

#### 6.3.2 Attention U-Net

Attention U-Net, first introduced in 2018 in [56], is a variation of the U-Net model that adds an Attention Gate to the original U-net model. The objective of this variant is to allow the model to automatically focus on relevant regions of the image by reducing the computational resources wasted on irrelevant activations, therefore providing better generalization for the network. The specific model implemented incorporates 9 344 841 parameters.

Attention Gates leverage the concept of soft attention, in which relevance is distributed on an image through a weight affectation, more relevant pixels or areas will have larger weights than irrelevant regions. These weights are automatically adjusted during training through backpropagation in order to determine the areas of interest.

The skip connections present in U-Net combine the spacial information present in the down-sampling path with the up-sampling path to retain good spacial information in the model, nevertheless this process brings along the poor feature representation from the initial layers. Soft-attention gates implemented at these skip connections aim to incorporate this spacial information while maintaining the rich feature representation from the up-sampling layers. In Figure 23 the Attention U-Net model structure is presented and in Figure 24 the decomposition of an Attention Gate.

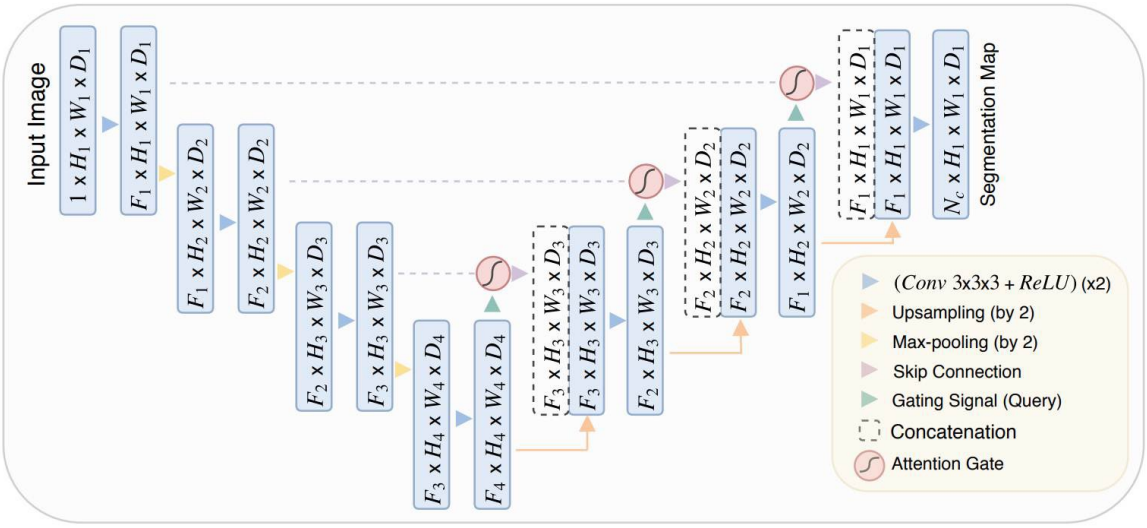



Figure 23: Attention U-Net [56]

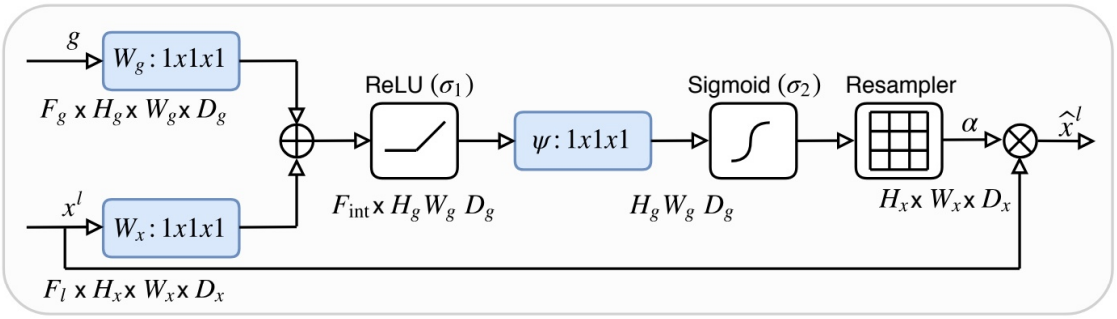



Figure 24: Attention Gate [56]

An attention gate takes two inputs, $g$ and $x$. $g$ corresponds to the gating signal, that comes from the up-sampling layer under the attention gate, and therefore has better feature representation. $x$ comes from skip connections (early layers) and has better spacial information. The gate applies convolutions to both signals, resizes them and then adds them together. Through this addition, aligned weights get larger while unaligned weights get smaller, thus greater weights will correspond to well positioned regions which have strong feature and are therefore more relevant. Then a ReLu function is applied, a convolution and a sigmoid too, to make sure that the attention factors $\alpha \in [0, 1]$. Finally, an up-sampling is performed, to attain the original size of $x$ so it is possible to make an element-wise multiplication between $x$ and $\alpha$, which scales the vector $x$ based on relevance. These computations can be summarized as follows:

$$\begin{cases} q_{\text{att}} = \psi^T \sigma_1 \left( W_x^T x^l + W_g^T g_i + b_g \right) + b_\psi \\ \alpha = \sigma_2 \left( q_{\text{att}}(x^l, g_i; \Theta_{\text{att}}) \right) \\ \hat{x}^l = x^l \cdot \alpha \end{cases} \tag{1}$$

where

$$\sigma_2(x) = \frac{1}{1 + \exp(-x_{i,c})} \tag{2}$$

corresponds to the sigmoid activation function.

### 6.3.3 R2 Attention U-Net

R2U-Net is another variation of U-Net, also introduced in 2018 in [57], that was meant to improve the segmentation of medical images. The model implemented in this project incorporates the advantages present in the previously explained model (Attention U-net) and the ones present in R2 U-Net. Therefore, it has been called R2 Attention U-net.

R2U-Net stands for Residual Recurrent U-Net. It substitutes all conventional convolution blocks in U-Net with Residual Recurrent Convolution blocks. The difference between these blocks is detailed in the following Figure:

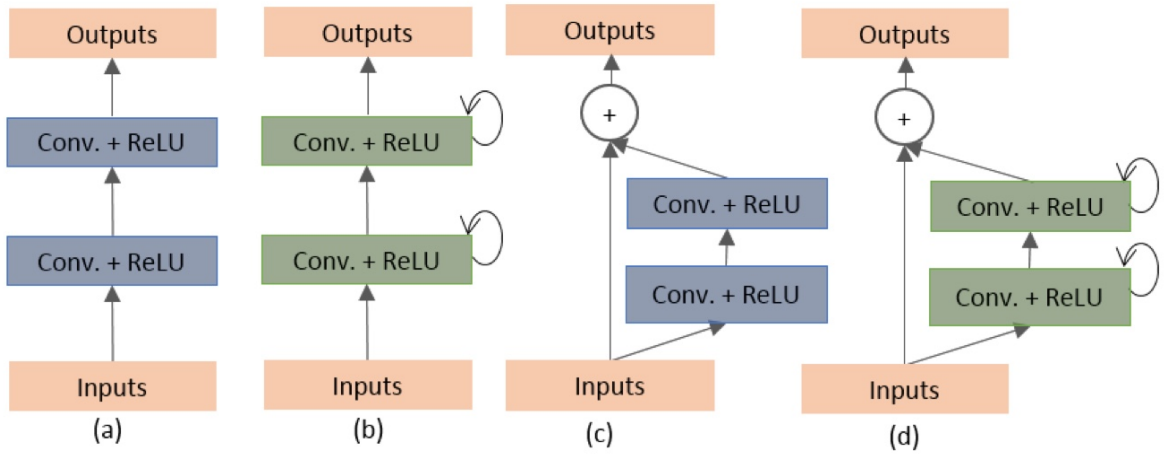



Figure 25: Different variant of convolutional and recurrent convolutional units (a) Forward convolutional units, (b) Recurrent convolutional block (c) Residual convolutional unit, and (d) Recurrent Residual convolutional units (RRCU) [57].

Recurrent Convolutional Layers (RCL), depicted in Figure 25(b) incorporate both a standard and a recurrent layer to combine the benefits of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to handle spatial and temporal dependencies respectively. The standard convolutional layer in an RCL processes the spatial features of the input data. The input to the standard convolutional layer at each time step is $x_l^{(i,j)}(t)$ which is the current state of the feature map that the layer is processing. The input to the recurrent layer at each time step $(t1)$ is denoted as $x_l^{r(i,j)}(t-1)$. This allows the layer to incorporate information from previous time steps into its current processing, similar to how RNNs operate. The output of the RCL at time step $t$ is a combination of the convolution applied to the current input and the recurrent connection from the previous state. The formulation can be represented as:

$$O_{ijk}^l(t) = (w_k^f)^T * x_l^{(i,j)}(t) + (w_k^r)^T * x_l^{r(i,j)}(t-1) + b_k$$

where

- $(w_k^f)^T * x_l^{(i,j)}(t)$ is the convolutional operation on the current input
- $(w_k^r)^T * x_l^{r(i,j)}(t-1)$ is the recurrent operation incorporating the previous state
- $b_k$ is the bias term

The output of the RCL is passed through a ReLU activation function, expressed as:

$$\mathcal{F}(x_l, w_l) = f(O_{ijk}^l(t)) = \max(0, O_{ijk}^l(t))$$

where $\mathcal{F}(x_l, w_l)$ represents the activated output.

For an RRCU, depicted on Figure 25d), the output $x_{l+1}$ at a subsequent time step $t+1$, is calculated as:

$$x_{l+1} = x_l + \mathcal{F}(x_l, w_l)$$

## 6.4 Results and analysis

### 6.4.1 Classification

Due to the complexity of camouflaged animal images and the wide variety of species present in the dataset provided by CS Group, the segmentation algorithm intended to be developed in this project will begin with a classification of the image to be segmented. Based on this classification, the image will be input into a segmentation model (U-Net or its variants) that has been trained with images of that specific class. The goal of developing such a method is to address the significant peculiarities and differences present in images of various animals by creating more specialized models capable of recognizing the specific characteristics of each animal class. Specifically, this project fully develops the classification algorithm and trains and studies a segmentation algorithm prepared with images of terrestrial animals (dogs and cats). However, the results obtained can be easily extrapolated to other types of animals belonging to different classes.

The training of the classification model was initiated for 10 epochs but was stopped prematurely at the 6th epoch due to early stopping. Configured with a patience of 5 epochs, this indicates that the model did not show any further improvement after the first epoch. Once the model was trained, the model was able to make predictions, which allowed the generation of the following confusion matrices:

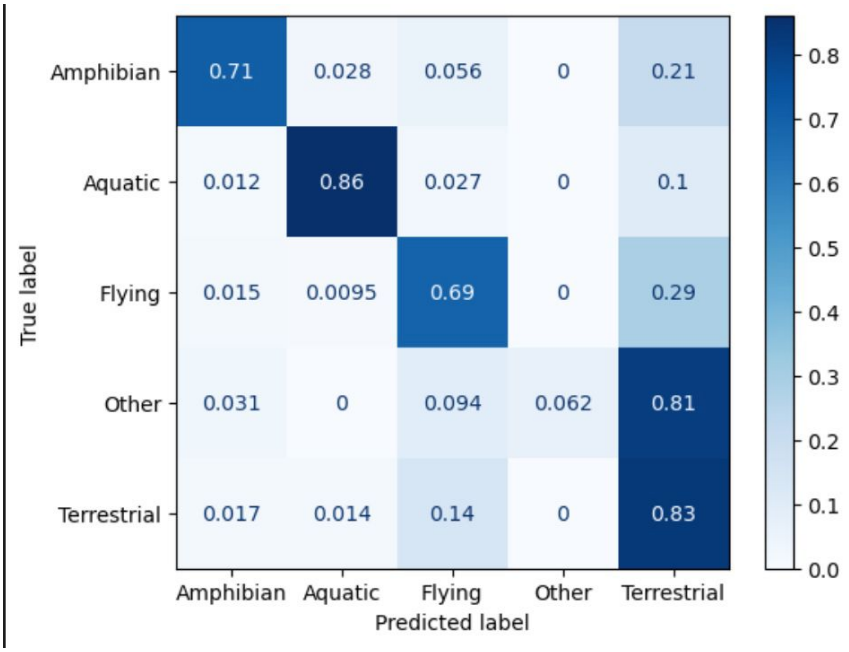



Figure 26: Confusion Matrix - Training Set

By analyzing the confusion matrix obtained from the training set predictions, one can observe that a certain number of images from the "Amphibian" and "Flying" classes are

misclassified as "Terrestrial". This is not surprising, given that these two environments share characteristics with terrestrial environments. Additionally, the "Other" class is classified as "Terrestrial" in 81% of the cases. Upon examining the images in the "Other" class, it can be observed that the majority of these images could belong to other categories, particularly "Terrestrial". It is likely that these images were placed in this category by the dataset creator because they were not intended for use. Therefore, the results are consistent with this hypothesis.

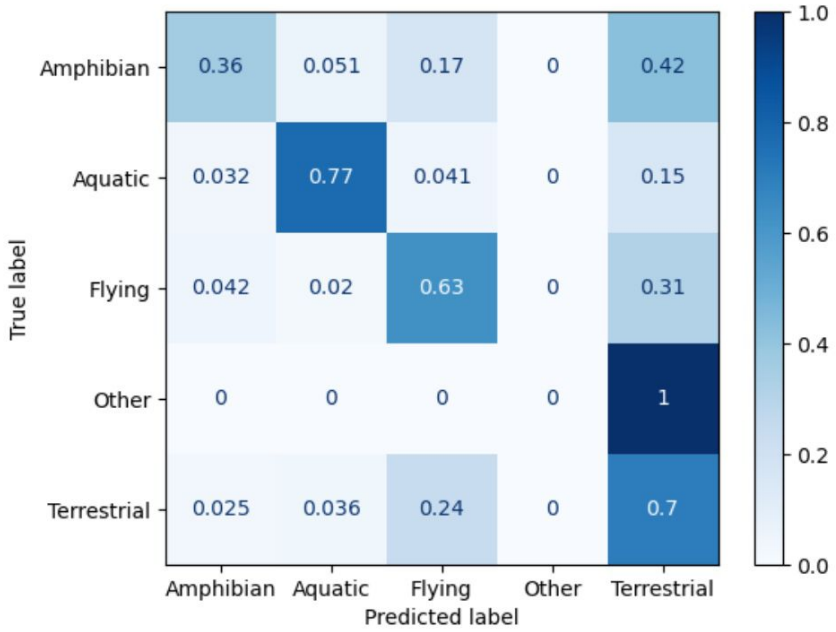



Figure 27: Confusion Matrix - Validation Set

In the confusion matrix obtained from the validation set predictions, one can notice the same misclassifications but in a more pronounced manner.

- Only 36% of "Amphibian" images are correctly classified, with 42% misclassified as "Terrestrial"
- All images in the "Other" category are misclassified as "Terrestrial"

These mixed performances are probably also due to the reasons mentioned earlier. In any case, it is evident the classification model has difficulty generalizing.

In the end, an accuracy of 66% on the validation set was attained. However, this performance could likely be improved by removing the "Other" class or redistributing its images into other categories.

#### 6.4.2 Segmentation Training

Each model (U-Net, Attention U-Net and R2 Attention U-Net) was trained on both:

- the Non Camouflaged Dataset ($D_{NONCAM}$)
- the Mixed Dataset ($D_{BOTH}$)

In order to study their performance and compare the results. First and foremost, let's look at how the models behaved during training. The following Table summarizes some important information about training:

| Model | Number of parameters | Dataset | Training Time (h:min:s) | Number of epochs |
|---|---|---|---|---|
| UNET | 7,771,873 | D_NONCAM | 06:57:34 | 12 |
| | | D_BOTH | 01:04:44 | 12 |
| Attention | 9,344,841 | D_NONCAM | 17:15:33 | 20 |
| | | D_BOTH | 02:41:39 | 20 |
| Residual | 9,787,497 | D_NONCAM | 18:50:35 | 20 |
| | | D_BOTH | 03:04:55 | 20 |

For each dataset used to train the models, 80% of the data was used for training and the remaining 20% for validation. This represents:

- For $D_{NONCAM}$: 5905 images for training and 1477 for validation
- For $D_{BOTH}$: 960 images for training and 241 for validation

The significant difference in training times between models trained on $D_{NONCAM}$ and $D_{BOTH}$ datasets is primarily due to the larger size of $D_{NONCAM}$. On top of that, Attention U-Net and R2 Attention U-Net have an increased number of parameters compared to the basic U-Net. Therefore, the added complexity from attention mechanisms and recurrent layers requires longer training time. It is also important to note that all models where supposed to be trained for 20 epochs but U-Net stopped early, as the validation loss didn't improve.

The following figures depict the evolution of the training and validation loss during the training of the different models:

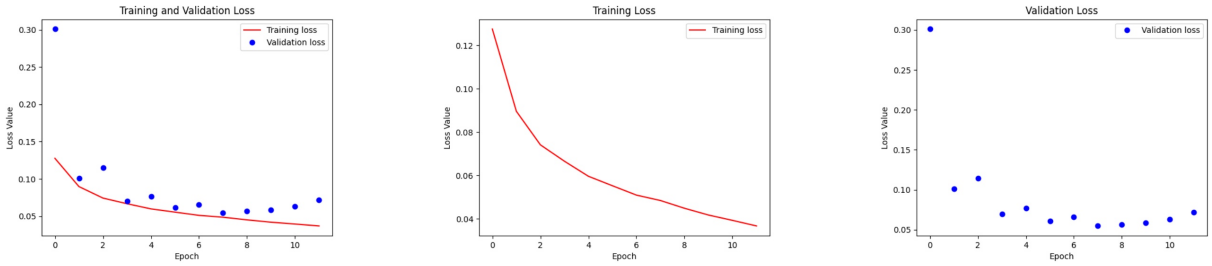



Figure 28: U-Net trained on $D_{NONCAM}$

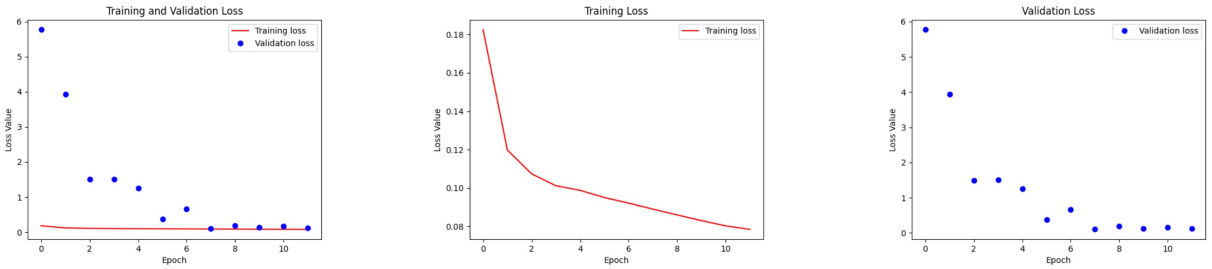



Figure 29: U-Net trained on $D_{BOTH}$

As it can be seen, The training and validation loss curves for U-Net indicates a steady decline, suggesting effective learning. However, the model exhibits some overfitting towards the end of training. Notably, the training process for U-Net stopped early due to the early stopping mechanism, as the validation loss stopped improving before the predefined number of epochs was reached. This early stopping indicates that while U-Net learns reasonably well, it hits a performance ceiling earlier compared to the more advanced models. It is also important to note the difference in scale between the loss

values from the training on $D_{NONCAM}$ and $D_{BOTH}$: for $D_{NONCAM}$, initial values for the validation loss under 0.3 and for $D_{BOTH}$ under 6 are observed. This must be taken into account when deducing conclusions from the charts. This difference will still be present for Attention U-Net and R2 Attention U-Net.

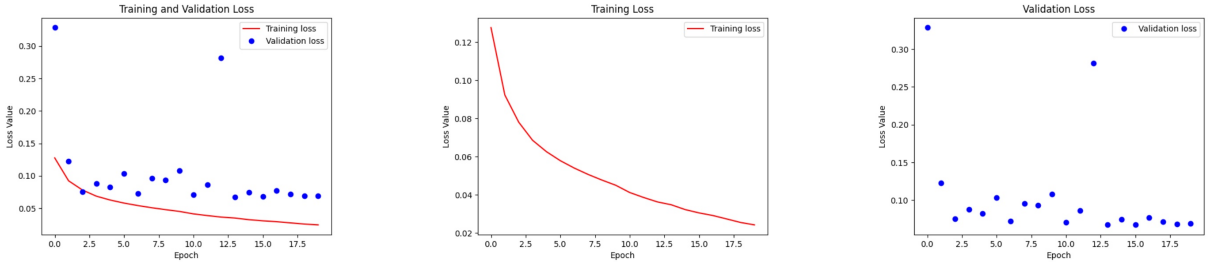



Figure 30: Attention U-Net trained on $D_{NONCAM}$

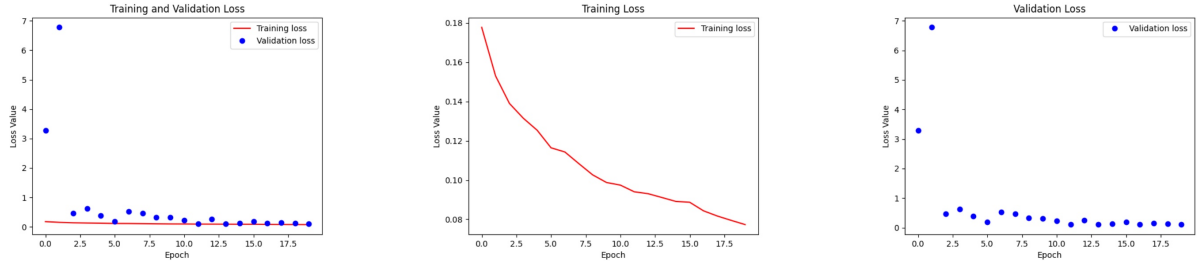



Figure 31: Attention U-Net trained on $D_{BOTH}$

For Attention U-Net, the loss curves show a consistent decline for both training and validation sets, despite a huge increase in validation loss after the 13th epoch when training on $D_{NONCAM}$. This loss increase is likely due to the stochastic nature of the Adam compiler which can cause the loss to increase momentarily, although it ensures convergence towards a minima (very similar to SGD). The loss decline indicates good generalization and effective learning. The attention mechanisms integrated into the model allow it to focus on relevant features, explaining the continuous improvement of segmentation performance. Unlike U-Net, Attention U-Net did not trigger early stopping, suggesting that the model could continue to learn and improve even at the end of the training period. Therefore, a longer training, not feasible in this case due to time restrictions, would probably lead to better results.

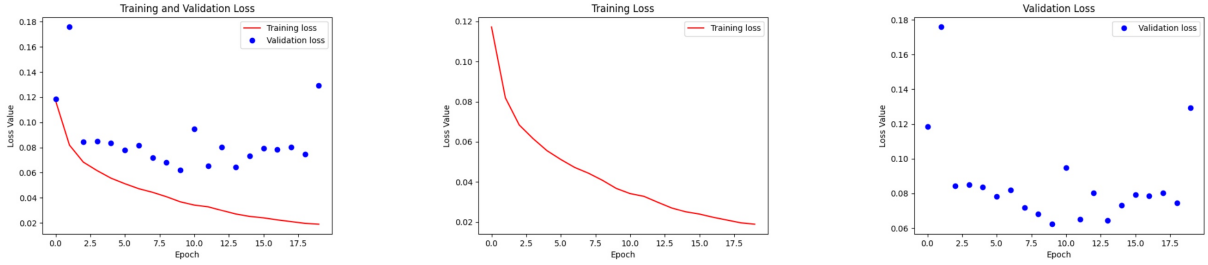



Figure 32: R2 Attention U-Net trained on $D_{NONCAM}$

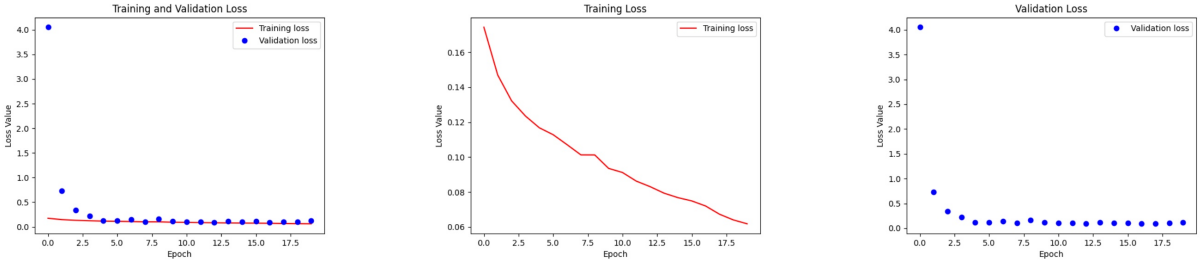



Figure 33: R2 Attention U-Net trained on $D_{BOTH}$

For R2 Attention U-Net, the training and validation losses decrease steadily with minimal fluctuations, indicating robust learning and strong generalization capabilities. The recurrent and residual blocks enhance the model's ability to maintain context and improve learning stability. Similar to Attention U-Net, R2 Attention U-Net did not activate the early stopping mechanism, suggesting continuous improvement throughout the training process. Just like for Attenntion U-Net, more training time would very probably lead to better results.

### 6.4.3 Segmentation Performance

In this section the predictions computed by each of the three segmentation algorithms will be analyzed, comparing their results and their ability to segment both camouflaged and non-camouflaged animals. To understand the segmenting accuracy of the models, Intersection over Union (IoU) will be used as the metric of performance:

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \tag{3}$$

IoU is defined as the ratio of the intersection area of the predicted segmentation and the ground truth segmentation (overlap), to the union area of the predicted segmentation and the ground truth segmentation (total area covered by both). It can be interpreted as follows:

- IoU = 0: No overlap between the predicted segmentation and the ground truth.
- IoU = 1: Perfect overlap between the predicted segmentation and the ground truth.
- $0 <$ IoU $< 1$ : Partial overlap, with higher values indicating better segmentation accuracy.

As described in the previous section, each model has been trained on two different datasets: $D_{NONCAM}$, that only includes non-camouflaged images, and $D_{BOTH}$, that includes both camouflaged and non-camouflaged images but is notably less exhaustive. Which gives 6 different models.

Each model trained on $D_{NONCAM}$ has predicted:

- All the CAM Dataset
- The first 200 images of the training dataset ($D_{NONCAMtraining}$)
- All the validation dataset ($D_{NONCAMvalidation}$)

On the other hand, each model trained on $D_{BOTH}$ has predicted:

- All the images in the training dataset ($D_{BOTHtraining}$)
- All the images in the validation dataset ($D_{BOTHvalidation}$)

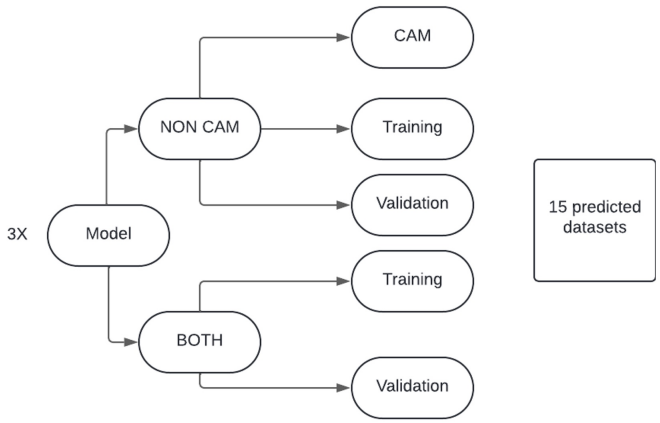



Figure 34: Prediction structure

The analysis performed on each prediction with respect to its ground truth through the IoU metric are summarized in the following table:

| Model | Number of parameters | Training Dataset | Evaluation Dataset | IoU |
|---|---|---|---|---|
| **UNET** | 7,771,873 | D_NONCAM | CAM | 0.452 |
| | | | Training Set | 0.866 |
| | | | Validation Set | 0.832 |
| | | D_BOTH | Training Set | 0.646 |
| | | | Validation Set | 0.637 |
| **Attention** | 9,344,841 | D_NONCAM | CAM | 0.508 |
| | | | Training Set | 0.865 |
| | | | Validation Set | 0.831 |
| | | D_BOTH | Training Set | 0.592 |
| | | | Validation Set | 0.590 |
| **Residual** | 9,787,497 | D_NONCAM | CAM | 0.520 |
| | | | Training Set | 0.808 |
| | | | Validation Set | 0.757 |
| | | D_BOTH | Training Set | 0.559 |
| | | | Validation Set | 0.532 |

Given the project's focus on camouflaged images and the generalization capabilities of the models, the analysis will be performed on:

- Models trained on $D_{NONCAM}$ when predicting on the CAM dataset of camouflaged dogs and cats
- Models trained on $D_{BOTH}$ on the validation set

The results can be reorganized in the following table:

| Training Dataset | Evaluation Dataset | Model | IoU |
|---|---|---|---|
| D_NONCAM | CAM | UNET | 0.452 |
| | | Attention | 0.508 |
| | | Residual | 0.520 |
| D_BOTH | Validation Set | UNET | 0.637 |
| | | Attention | 0.590 |
| | | Residual | 0.532 |

Among the models trained on $D_{NONCAM}$:

- The model that generalizes the best is R2 Attention U-Net (IoU = 0,520)
- The second-best is Attention U-Net (IoU = 0,508)
- The least performant is U-Net (IoU = 0,452)

However, among the models trained on $D_{BOTH}$:

- The model that generalizes the best is U-Net (IoU = 0,637)
- The second-best is Attention U-Net (IoU = 0,590)
- The least performant is R2 Attention U-Net (IoU = 0,532)

These results indicate that the models perform very well on non-camouflaged images but have a harder time segmenting camouflaged ones. There could be several reasons for this:

- The images present in the camouflaged dataset are too complex or non representative, given the small sizes of the animals on the images or the fact that only a specific part of the animal is sometimes present (for example its leg as in Figure 34).

Figure 35: Non-representative image from the CAM dataset

- The models need more camouflaged images to learn the features necessaries to segment them. We have already mentioned the scarcity of camouflaged datasets in the industry .
- The models are unable to recognize correctly camouflaged animals due to their structure or parameters. For example, a bigger kernel in the convolutive layers

can lead to a better recognition of textures and therefore potentially a better segmentation. To determine if better performances are achievable more study is needed on the subject.

Even though the performance is not impressive, the models are still able to recognize the camouflaged animals in many cases, as showcased in Figure 38. Some other segmentations are presented in Figures 36 and 37. Also, the fact that R2 Attention U-Net is able to generalize on camouflaged images when trained on a non-camouflaged dataset, while still having significant potential for improvement through extended training, appears promising.

#### 6.4.4 Segmentation Illustrations

To better illustrate the performance of the models, visual examples of the segmentation results are presented next to their original image (input) and the mask used for training.

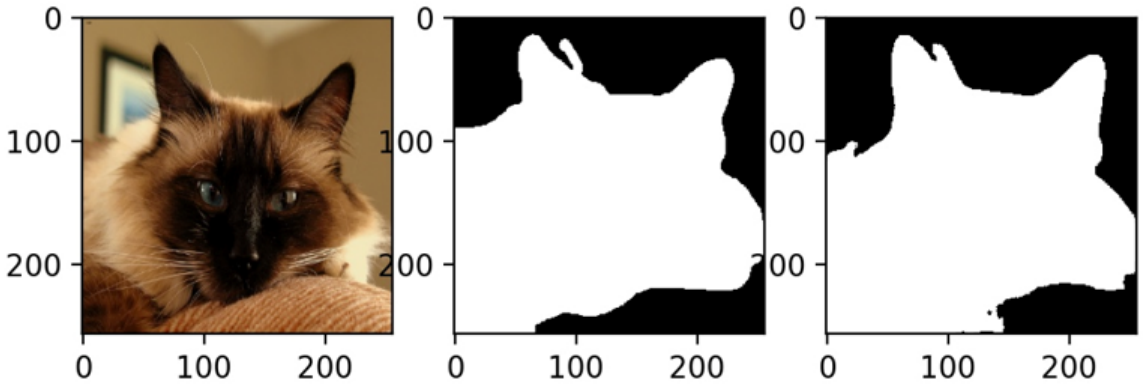



Figure 36: U-Net trained on $D_{NONCAM}$ - Segmentation of a non camouflaged image from the validation set

In this example, the non-camouflaged cat is well-segmented. This level of segmentation quality is consistent across the other models that were trained (Attention U-Net and R2 Attention U-Net) when segmenting non camouflaged animals.

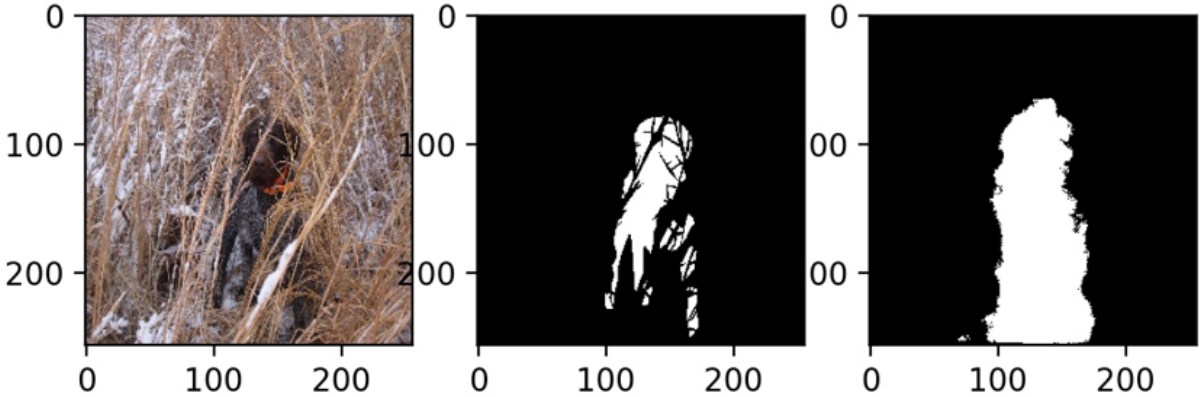



Figure 37: U-Net trained on $D_{both}$ - Segmentation of a camouflaged image from the validation set

Here's an example of a camouflaged animal with good segmentation, but as it can be seen it isn't perfect. The U-Net model was able to capture the area in which the animal was present, but unable to determine which details belonged to the dog or to the background.

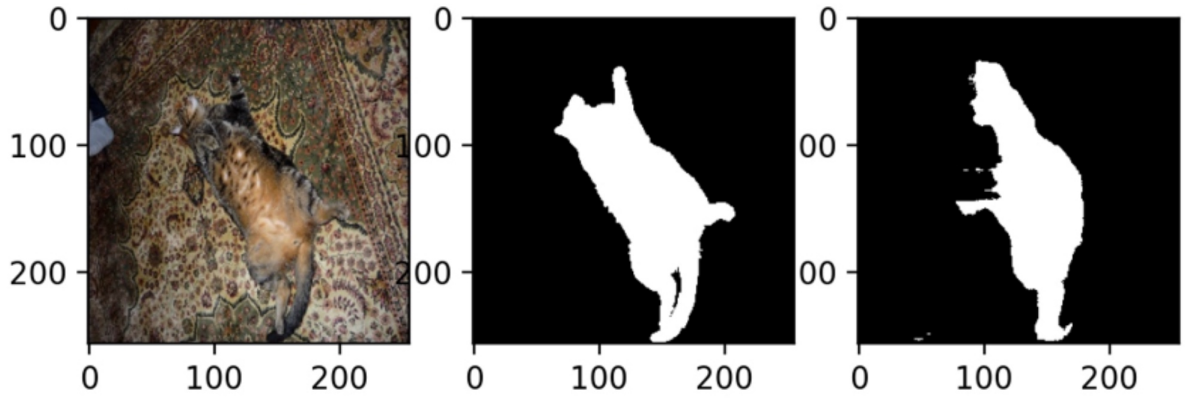


Figure 38: R2 Attention U-Net trained on $D_{NONCAM}$ - Segmentation of a camouflaged image from the camouflages dataset (CAMO)

Here's another example: a perfectly camouflaged cat in its environment. While R2 Attention U-Net successfully segmented the cat from the background, there's still room for improvement on the prediction's shape. As previously mentioned, a bigger kernel in the convolutional layers might ameliorate the network's ability to detect complex patterns.

### 6.5 Economic cost and Environmental Impact of The Models Training

The computer used for this project is provided by the university CentraleSupelec. It is specifically designed for the tasks required in this project, particularly the training of complex neural network models. This high-performance machine is equipped with four NVIDIA A100 GPUs (each with 80 GB of memory), a single AMD 7742 CPU (64 cores, 2.25 GHz base, 3.4 GHz boost), 512 GB of DDR4 RAM, a 7.68 TB U.2 NVMe drive for data storage, and a 1.92 TB M.2 NVMe drive for the operating system. The cost of building such a computer can range from 55,000 € to 80,000 €, depending on the market prices of components. This investment is necessary to handle the computational demands of training complex neural network models. However, it is important to note that this resource is shared among numerous students who also require significant computational power for their own projects. Therefore, we will not have access to the computer at all times, as it must be allocated among various users. This is why it will be assumed that the system was utilized only at 30% of its power during training.

The A100 GPU has a TDP (Thermal Design Power) of around 250W, and the AMD 7742 CPU has a TDP of 225W. Factoring in the other components, the total power consumption of the system could be around 550W under full load. At 30% usage, this equates to approximately 165W. Over 49 hours and 55 minutes, the total time employed on training all models, this results in around 8.25 kWh of electricity consumption. Given France's carbon intensity of 56.02 g CO/kWh (according to Statista.com in 2023 in France), this training session generated approximately 462.2 grams of CO.

In terms of cost, at an average electricity price of 0.047 € per kWh (according to Statista.com in June 2024 in France), the electricity cost for the training session is less than 0.4 €. Therefore, the total cost estimate for this training session includes the depreciated cost of the hardware, which could be prorated based on the portion of time used, and the minimal electricity cost, giving a rough estimate of under 0.4 € for electricity and a small fraction of the total hardware cost for a single training session.

While the financial costs associated with training AI models on the described computer setup are relatively low, they pale in comparison to the expenses incurred when training much larger models like GPT-4 or similar large-scale language models. The energy consumption for training on this setup is minimal, with a carbon footprint of 462.2 grams of CO and electricity expenses under 0.4 €, along with a small portion of the hardware's total value contributing to the overall cost. However, training massive models like GPT-4 involves an entirely different scale of resources. These models require thousands of high-end GPUs operating in parallel, specialized infrastructure, and weeks or even months of continuous operation. The energy consumption alone can reach into the hundreds of megawatt-hours, with CO emissions potentially reaching into the tens of thousands of kilograms and costs easily running into millions of euros for both the hardware and the electricity. Additionally, such large-scale training efforts necessitate significant investment in data storage, cooling systems, and ongoing maintenance, all of which add to the overall cost and environmental impact.

### 6.6 Improvement Avenues

While simpler CNN architectures like U-Net, Attention U-Net, and R2 Attention U-Net show potential for the segmentation of camouflaged animals, the results are not yet satisfactory for practical applications. Fortunately, several avenues for improvement can be considered:

- **Enhanced Training of the Models:** Both Attention U-Net and R2 Attention U-Net can still benefit from extended training to improve their performance. Even after utilizing a full A100 GPU for 24 hours, the models were still showing improvements without overfitting, thanks to early stopping mechanisms. However, due to time constraints, the training had to be stopped. The results for R2 Attention U-Net were particularly promising. Utilizing more powerful computational resources to further train the models could potentially yield better results.
- **Exploring New Architectures:** Other CNN architectures beyond U-Net could be explored specifically for the detection of camouflaged animals. Investigating novel or hybrid architectures like MirrorNet [8] might provide better feature extraction and segmentation capabilities tailored to the challenges of camouflaged animal detection.
- **Refining the CAMO Dataset:** Various modifications can be made to the CAMO dataset. Some camouflaged animals in many images are really hard to detect even for the human eye. Removing images that are "too difficult" might be beneficial, as these could be hindering the learning process. Additionally, many camouflaged animals are very small in terms of pixel count in numerous images. One approach could be to detect images where animals are small (by counting the number of white pixels on the masks) and zoom in on these images to better visualize the animal.
- **Data Augmentation and Synthetic Data**: Further augmenting the existing dataset with synthetic data or applying more advanced augmentation techniques could help in creating a more robust model. Techniques such as GANs (Generative Adversarial Networks) [58] could be used to generate realistic camouflaged animal images, increasing the diversity and size of the training set.

- **Transfer Learning and Fine-Tuning**: Applying transfer learning from models pre-trained on large and diverse datasets can help in improving the performance on our specific task. Fine-tuning these models on the dataset could leverage the learned features and improve segmentation accuracy.
- **Integration of Multimodal Data**: Incorporating additional data modalities, such as thermal imaging or depth information, could provide complementary information to improve segmentation accuracy. This multimodal approach could help in distinguishing camouflaged animals more effectively.

Exploring these improvement strategies may enhance the effectiveness of the models in segmenting camouflaged animals and contribute to the development of more robust and reliable detection systems.

## Conclusion

Artificial intelligence (AI) is rapidly emerging as a transformative technology that will undoubtedly reshape our future. Understanding its complexity, economic impact, potential shifts in the labor market, and environmental footprint is of paramount importance. This project, an extension of a study performed in collaboration of CS Group, explores these critical issues by examining the economic and environmental impacts of AI as a whole, and by tackling a specific challenge —camouflaged animal segmentation— to delve into the intricacies associated with developing sophisticated AI models.

A key focus of this project was to evaluate how well simpler Convolutional Neural Network (CNN) architectures, including U-Net, Attention U-Net, and R2 Attention U-Net, could generalize and effectively segment camouflaged animals. The findings reveal the complexity involved in developing AI algorithms capable of addressing such challenging tasks. While these models demonstrated high accuracy in segmenting non-camouflaged animals, achieving strong performance metrics and confirming their robustness for standard segmentation tasks, they struggled significantly when applied to camouflaged images. The results were less satisfactory, even for the best-performing models, underscoring the difficulty of accurately segmenting camouflaged animals. This challenge highlights the inherent complexity of designing AI systems that can handle more nuanced and demanding real-world scenarios.

Despite these challenges, there is potential for improvement. R2 Attention U-Net, in particular, showed promising results when trained on non-camouflaged images ($D_{NONCAM}$) and evaluated on camouflaged images, suggesting that more sophisticated architectures and enhanced computational resources could lead to better performance. Extended training and the application of more powerful models could further refine these results, pointing to the need for continuous advancement in AI development to tackle increasingly complex tasks.

On the economic front, AI is poised to significantly boost productivity across various sectors, potentially leading to substantial economic growth. However, the development and widespread adoption of this technology have also sparked concerns about its impact on labor demand. As detailed in this study, AI is likely to influence jobs that require higher education and offer higher wages, contributing to a reduction in income inequality between the 90th and 10th percentiles. However, it may also exacerbate disparities

between the top 1% and the rest, particularly between the 99th and 90th percentiles. These shifts suggest that while AI holds promise for economic advancement, it also presents challenges that need careful consideration, particularly in terms of workforce displacement and income distribution.

From an environmental perspective, the increasing computational demands required to develop new AI algorithms are projected to exacerbate the environmental impact of AI. The energy consumption associated with training complex models, coupled with the carbon emissions from data centers, poses a growing concern. The project underscores the difficulty in assessing the specific environmental impact of an AI model throughout its lifecycle, a challenge that has not been extensively addressed in existing literature. To fill this gap, several methodologies from current research are proposed within this project, offering a framework for better understanding and mitigating the environmental costs associated with AI.

Therefore AI is a powerful and fast-growing technology with the potential to drive significant economic growth and innovation. However, it also brings with it substantial challenges, particularly in terms of labor market shifts, environmental sustainability, and the complexity of developing highly performant algorithms capable of addressing difficult tasks such as camouflaged animal segmentation. As AI continues to evolve, it is crucial to balance its benefits with a thoughtful approach to its economic and environmental impacts. This project contributes to this ongoing discourse by providing insights into the complexities of AI development and proposing methodologies to better understand and address the associated challenges.

## References


[1] CHARLES S. WEAVER. "Some Properties of Threshold Logic Unit Pattern Recognition Networks". https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1672802 (1975).

[2] Rosenblatt, F. "The perceptron: A probabilistic model for information storage and organization in the brain". Psychological Review, 65(6), 386–408. https://doi.org/10.1037/h0042519 (1958).

[3] Aurélien Géron, "Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow", Book, O'REILLY, (2019).

[4] Fatemeh Farnaghizadeh, "Semantic_Segmentation_U-NET"https://www.kaggle.com/code/fatemehfarnaghizadeh/semantic-segmentation-u-net (November 2023).

[5] Bing Xu et al., "Empirical Evaluation of Rectified Activations in Convolutional Network," https://arxiv.org/pdf/1505.00853 (2015).

[6] Djork-Arné Clevert et al., "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)" Proceedings of the International Conference on Learning Representations, https://arxiv.org/pdf/1511.07289 (2016).

[7] Günter Klambauer et al., "Self-Normalizing Neural Networks," Proceedings of the 31st International Conference on Neural Information Processing Systems, https://arxiv.org/pdf/1706.02515 (2017).

[8] Jinnan Yan, Trung-Nghia Le, Khanh-Duy Nguyen, Minh-Triet Tran, Thanh-Toan Do, and Tam V. Nguyen, "MirrorNet: Bio-Inspired Camouflaged Object Segmentation", https://arxiv.org/pdf/2007.12881 (2021).

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation", in IEEE Conference on Computer Vision and Pattern Recognition, https://arxiv.org/pdf/1411.4038 (June 2015).

[10] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge", International Journal of Computer Vision, vol. 88, no. 2, pp. 303–338.

[11] J. Shotton, J. M. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," International Journal of Computer Vision, vol. 81, no. 1, pp. 2–23 (2009).

[12] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards realtime object detection with region proposal networks", in Conference on Neural Information Processing Systems, pp. 91–99, https://arxiv.org/pdf/1506.01497 (2015)

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Conference on Computer Vision and Pattern Recognition, https://openaccess.thecvf.com/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf (2016).

[14] Trung-Nghia Lea, Tam V. Nguyenb, Zhongliang Nieb, Minh-Triet Tranc, Akihiro Sugimotod, "Anabranch Network for Camouflaged Object Segmentation", https://arxiv.org/pdf/2105.09451v1 (2021).

[15] Christen, Peter et al. “A Review of the F-Measure: Its History, Properties, Criticism, and Alternatives”, https://dl.acm.org/doi/pdf/10.1145/3606367 (2023).

[16] Zheng, Zhaohui, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye and Dongwei Ren. “Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression” https://arxiv.org/pdf/1911.08287 (2019).

[17] Etienne Delort, Laura Riou, Anukriti Srivastava. Environmental Impact of Artificial Intelligence: Bibliographic Report - Artificial Intelligence and Eco-responsibility Internship. INRIA; CEA Leti. 2023, pp.1-33. hal-04283245

[18] Hilty and Lorenz, ”Hilty ICT and Sustainability” Chapters 7, (Apr. 2008).

19 Johan Rockström et al. “A safe operating space for humanity”. In: Nature 461.7263, https://doi.org/10.1038/461472a (Sept. 2009).

[19] Katherine Richardson et al. “Earth beyond six of nine planetary boundaries”. In: Science Ad- vances 9.37, https://www.science.org/doi/10.1126/sciadv.adh2458 (2023).

[20] Charlotte Freitag, Mike Berners-Lee, Kelly Widdicks, Bran Knowles, Gordon Blair, and Adrian Friday, ”The climate impact of ICT: A review of estimates, trends and regulations”, https://arxiv.org/pdf/2102.02622 (2021).

[21] Etienne Delort, Laura Riou, Anukriti Srivastava, ”Environmental Impact of Artificial Intelligence”, https://inria.hal.science/hal-04283245. (2023).

[22] Statista, ”IoT devices installed base worldwide 2015-2025”, https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/ (2023).

[23] OpenAI, ”AI and compute”, https://openai.com/research/ai-and-compute, (May 2018).

[24] Emma Strubell, Ananya Ganesh, and Andrew McCallum. “Energy and Policy Considerations for Deep Learning in NLP”, http://arxiv.org/abs/1906.02243, (2019).

[25] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni, “Green AI”, https://dl.acm.org/doi/10.1145/3381831 ,(Nov. 17, 2020).

[26] Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau, ”Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning”, https://arxiv.org/pdf/2002.05651 ,(2022).

[27] Anne-Laure Ligozat and Sasha Luccioni, “A Practical Guide to Quantifying Carbon Emissions for Machine Learning researchers and practitioners”, https://hal.science/hal-03376391/document ,(2021).

[28] International Telecommunication Union, “Methodology for Environmental Life Cycle Assessments of Information and Communication Technology Goods, Networks and Services”, (Dec. 2014).

[29] Lasse F. Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan, “Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models”, https://arxiv.org/abs/2007.03051 , (2020).

[30] Lacoste Alexandre, Luccioni Alexandra, Schmidt Victor, and Dandres Thomas, "Quantifying the Carbon Emissions of Machine Learning", https://arxiv.org/pdf/1910.09700, (2019).

[31] Loic Lannelongue, Jason Grealey, and Michael Inouye. "Green Algorithms: Quantifying the Carbon Footprint of Computation", https://onlinelibrary.wiley.com/doi/abs/10.1002/advs.202100707, (2021).

[32] Sachin Mehta, Marjan Ghazvininejad, Srinivasan Iyer, Luke Zettlemoyer, and Hannaneh Ha-jishirzi, "DeLighT: Very Deep and Light-weight Transformer", https://arxiv.org/abs/2008.00623 (2021).

[33] Gemma Brady, Niraj Kapur, Jon Summers, and Harvey Thompson. "A case study and critical assessment in calculating power usage effectiveness for a data centre", https://www.sciencedirect.com/science/article/abs/pii/S0196890413004068 (Dec. 2013).

[34] A.S. Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. "Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model", https://arxiv.org/abs/2211.02001 (2022).

[35] Ledger-Jessop, K, "Ethical Considerations in AI Implementation: Challenges and Opportunities", https://dx.doi.org/10.3351/ppp.2022.5954874746 (2022).

[36] Partridge, M. D., Tsvetkova, A., and Betz, M. R., "Economic Growth, Job Growth, and Income Inequality: A Regional Perspective", https://dx.doi.org/10.1111/jors.12499 (2020).

[37] Zhaoxia Yi, Shirley Ayangbah, "The Impact of AI Innovation Management on Organizational Productivity and Economic Growth: An Analytical Study", https://www.ijbmer.org/uploads2024/BMER_7_580.pdf (2024).

[38] Kim, S., Lee, S., and Kim, Y., "The Impact of AI on Productivity in the Manufacturing Sector", https://dx.doi.org/10.1080/10168737.2021.1952641 (2021).

[39] Carneiro, N., Figueira, G., and Costa, M., "A data mining based system for credit-card fraud detection in e-tail", (2018).

[40] Criscuolo, C., Gal, P. N., and Menon, C., "Productivity Growth and Inclusive Growth: An International Perspective", https://dx.doi.org/10.1787/52ab4e26-en (2020).

[41] Somashekhar, S. P., Sepúlveda, M. J., Puglielli, S., Norden, A. D., Shortliffe, E. H., Rohit Kumar, C., ... and Ramya, Y., "Watson for oncology and breast cancer treatment recommendations: Agreement with an expert multidisciplinary tumor board", https://pubmed.ncbi.nlm.nih.gov/29324970/ (2018).

[42] Zhavoronkov, A., Ivanenkov, Y. A., Aliper, A., Veselov, M. S., Aladinskiy, V. A., Aladinskaya, A. V., ... and Aspuru-Guzik, A., "Deep learning enables rapid identification of potent DDR1 kinase inhibitors. Nature Biotechnology", (2019).

[43] McKinsey Global Survey on AI, https://www.mckinsey.com (2024).

[44] Brynjolfsson, E., and McAfee, A., "The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies", W.W. Norton and Company, (2020).

[45] Esfahani, H. S., Rezai, A., and Steger, T. M., "Economic Growth, Income Inequality, and Policy: An Introduction to the Special Issue", https://dx.doi.org/10.2139/ssrn.3554079 (2020).

[46] Seo, H. J., Choi, Y. S., and Lee, S., "Income Inequality and Economic Growth: The Role of AI", https://dx.doi.org/10.3390/su12145740 (2020).

[47] Hailemariam, A., & Dzhumashev, R., "The Impact of Technological Change on Income Inequality: A Quantitative Analysis", https://dx.doi.org/10.1515/SNDE-2018-0084 (2023).

[48] Romer, P. M., "Endogenous technological change. Journal of Political Economy", https://web.stanford.edu/~klenow/Romer_1990.pdf (1990).

[49] Michael Webb, "The Impact of Artificial Intelligence on the Labour Market", https://www.michaelwebb.co/webb_ai.pdf (2020).

[50] Acemoglu, Daron and Pascual Restrepo, "Robots and Jobs: Evidence from US Labor Markets", NBER Working Paper Series w23285, (2017).

[51] Honnibal, Matthew and Mark Johnson, "An Improved Non-monotonic Transition System for Dependency Parsing", EMNLP, 1373–1378., https://aclanthology.org/D15-1162.pdf (2015).

[52] Miller, George A., "WordNet: A Lexical Database for English", Communications of the ACM, 38(11): 39–41. (1955).

[53] Visual Geometry Group at Oxford, "Oxford-IIIT Pet Dataset", https://www.kaggle.com/datasets/tanlikesmath/the-oxfordiiit-pet-dataset

[54] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", https://arxiv.org/abs/1505.04597 (2015).

[55] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert, "Attention U-Net: Learning Where to Look for the Pancreas", https://arxiv.org/abs/1804.03999 (2018).

[56] Md Zahangir Alom1, Mahmudul Hasan, Chris Yakopcic, Tarek M. Taha, and Vijayan K. Asari1, "Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation", https://arxiv.org/abs/1802.06955 (2018).

[57] Radford, Alec et al., "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks" CoRR abs/1511.06434, https://arxiv.org/abs/1511.06434 (2015)