ADVANCED MODELS FOR COMPUTER VISION

Miguel Huertas Collado

Abstract - This project analyses and trains different computer vision algorithms, for image classification and action recognition in video. To do so, convolutional neural networks have been used, which are the state-of-the-art solution for image recognition, allowing the network to have very deep architectures. In terms of image classification, Faster-RCNN has been implemented for bounding box classification, validating the results with the widely-known YOLO algorithm. Moreover, U-Net has been trained for image segmentation, comparing both approaches and each model behaviour in terms of the parameters that shape the model. Finally, SlowFast Network has been trained for action recognition in video, where the algorithm has to focus in the context of the action, not only in the object and its localization, as in the previous case.

I. Introduction

Computer Vision (CV) is the field that studies and analyzes images and videos of the real world so that a computer can understand them, extracting and processing their information. The integration of Computer Vision in the industry, added to brand-new concepts as the Internet of Things (IoT), has reshaped the business into what is called the Industry 4.0. Modern computer vision systems are able to work in changing environments, far from the structured and repetitive tasks from years ago. They can interact with people, respond to interactions in different situations and work independently.

Today, CV algorithms incorporate convolutional neural networks to process this information. These networks are often named as black boxes, limiting their use in many cases, due to the lack of knowledge of what is happening inside them, making it difficult to treat them and the possibility of adapting them to different uses.

This project examines these algorithms, choosing actual models and training them 'from scratch'. By that, the model is analyzed in terms of its parameters, understanding its behavior for a potential integration in more complex applications.

Regarding images, Computer Vision addresses the problem of image processing classifying them in 4 different ways, which are collected visually in the following figure:

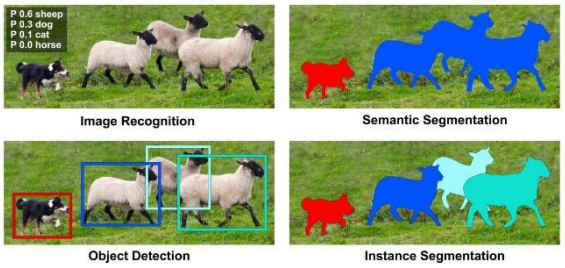

Figure 1: Image classification tasks.

- **Image classification**: identifies the object or the action of the picture.
- **Object detection:** each object is classified and localized by rectangular boxes around each object, known as bounding box.
- **Semantic segmentation**: each pixel of the image is labelled with a class.
- **Instance segmentation**: each pixel is classified in the same way as semantic segmentation,

but differencing between objects from the same class.

In terms of video classification, the task has been action recognition, which not only objects are recognized, but the context of the action needs to be analyzed as well.

II. Methodology

Convolutional neural networks have been used to address the image classification task in this project. The following algorithms have been trained:

1. **Faster-RCNN** [1] is an algorithm for bounding box object detection, which consists on a two-stage detector that shares the same convolutional neural network to accomplish both tasks: region proposal generation and object detection.
    a. Region Proposal Network (RPN): The input image goes through the convolutional network to obtain the feature maps. Then, a sliding window is applied for each location over the feature map, where k anchor boxes are used for each location to generate region proposals. Finally, a classification layer outputs 2k scores whether there is an object or not for every k boxes, while a regression layer outputs 4k scores for the coordinates of the box (x, y, width and height).
    b. Region of Interest pooling (RoI): it extracts fixed-sized feature maps for each proposal, reused from the existing convolutional feature map. Then, Max-Pooling is applied on every region. Hence, the output of RoI is always the same.
    c. Region-based Convolutional Neural network (R-CNN): two Fully Connected layers are added at the end: the first layer classifies the object while the second one process the boxes for each class.

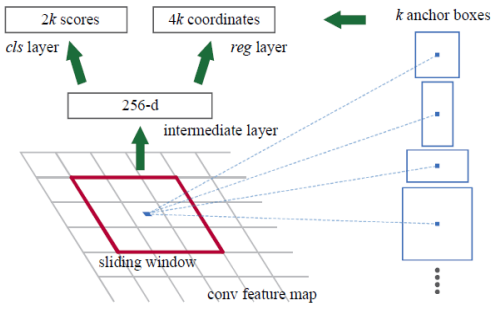

Figure 2: Region Proposal Network representation.

- **U-Net** [2]: Analyze the classification of images using semantic segmentation. U-Net consists on a Fully Convolutional Network whose inputs and outputs have the same size. It is divided into contraction and expansion path:
    - **Contraction path**: two 3x3 convolutions followed by a 2x2 maximum pooling. As it goes more advanced features are extracted and the size of the feature maps is reduced.
    - **Expansion path**: two 3x3 convolutions and 3x3 up-sampling convolutions to resize the image to its original size. By doing that, advanced features are obtained at the expense of localization information. To avoid that, lateral connections between feature maps from the contraction and the expansion paths are done.
    - **Overlap-tile strategy**: Padding is equal to zero in each convolution. Therefore,

the size of the output is smaller than the input. To solve that, overlap tile strategy is used to predict each part of the image dividing the input image in smaller parts. Next figure shows an example where the blue area is used to predict the yellow one. When the yellow box is close to the boundary, mirroring is applied to extrapolate the image.

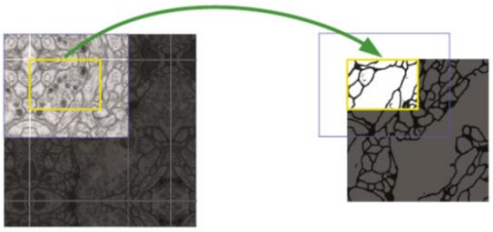

Figure 3: Example of overlap-tile strategy.

- **SlowFast Networks** [3]: It has been the algorithm chosen to perform action recognition in videos. Facebook AI Researchers analysed that in every video usually exists two distinct parts: dynamics areas where something important is going on and static areas, where frames change very slowly or do not even change. Therefore, Slow-Fast assigns a high definition Convolutional Neural Network to capture fast and fine motion (Fast pathway), and a low-definition CNN to analyse the spatial semantic content (Slow pathway). The compute cost of the slow pathway is 4 times higher than the cost of the Fast pathway.

  Both pathways use 3D ResNet models, but the difference resides in the stride of each path, being set at 2 sampled frames per second for the Slow path one and 15 sampled frames per second for the Fast pathway. In addition, the channel size of the paths differs also, being the channel size of the Fast path 1/8 of the Slow one. The advantage of doing that is reducing the computational requirements of the Fast path without compromising the performance.

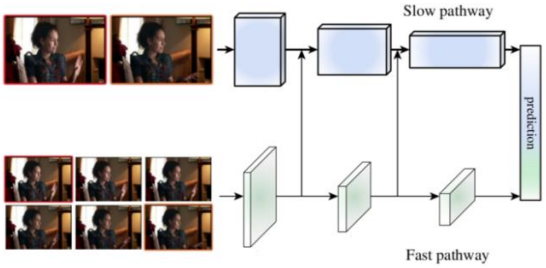

Figure 4: SlowFast Network structure.

As it was shown in the image above, data from the Fast path is fed into the Slow one through lateral connections to improve the performance of the model. Then, at the end of each path, Global Average Pooling is performed to reduce dimensionality. Finally, the result is introduced into a fully connected layer with a SoftMax to classify the action.

When training every model, a prior exploration of the state of the art has been carried out, considering scripts and the work already done that can serve as a starting point for the development of the project.

Training these models requires GPUs (Graphics Processing Units) to reduce execution time, being unsustainable training them only with a CPU. Therefore, Google Colaboratory, a cloud service adapted for Machine Learning, and a personal GPU have been used.

III. Results

1. Faster-RCNN

Due to the large number of parameters in the model (136 million), Google Colab has been used for training, except for the first simulation showed below (learning rate), where GPU NVIDIA GTX 2060 was used.

First, the model has been trained from scratch for 3 classes: 'people', 'traffic signs' and 'traffic lights'. Later, 'cars' and 'buses' classes were added by Transfer Learning, comparing both approaches using different parameters mentioned in next section. Finally, results have been validated with an already consolidated model in the state-of-the-art literature, such as YOLO (You Only Look Once).

The following configuration has been chosen as the starting point:

- 1904 training images, 479 images for validation and 388 for test.
- Image size: 150x150.
- Batch size: 1.
- Learning rate: $1x10^{-4}$.

Starting from the base conditions, the figure in the right column shows the path followed to optimize the model.

Therefore, the final learning rate has been $1x10^{-4}$, the batch size equals to 1 and the image size is 600x600, which corresponds to the green rectangles showed before. The model has obtained a precision value of 0.35 for testing, what means that when analyzing images that the model has never seen, the 35% of objects are classified correctly, locating them in the same way as a human would do. Next examples show the output of the model:

Figure 5: Structure followed during the project

Figure 6: Sample where class person is predicted

Figure 7: Class traffic sign is predicted twice with a 98% of confidence

Once the model is trained by means of *Transfer Learning*, considering as the starting point the best configuration previously shown, the model improves its precision up to 0.37. The images shown below are the result of this latest model:

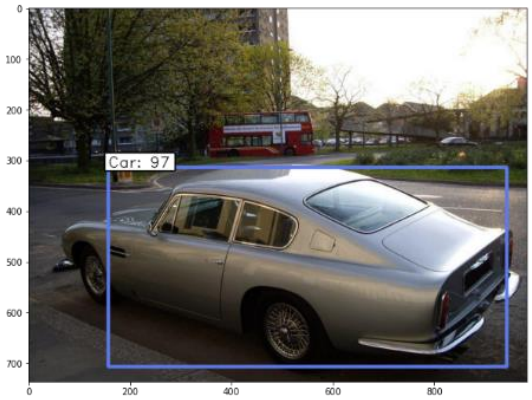

Figure 8: Class car is predicted with a 97% of confidence.

Figure 9: Class bus is predicted with a 97% of confidence.

2. U-Net

This model, less demanding than the previous one, allows training the entire model from scratch. However, it has been decided to train each class separately and integrate all together into the model, due to limited resources available.

Input parameters have been the following:

- 4000 training images, 800 images for validation and 800 for test.
- Image size: 128x128.
- Batch size: 2.
- Learning rate: $1x10^{-4}$.

Next figure shows the procedure for optimizing the model:

Figure 10: Structure followed during the project.

The final precision of the model, considering all classes, is 0.42. In this case, precision is performed considering each pixel in the image, whether it belongs to an object or not.

Next images are the output of the overall model:

Figure 11: Sample of class 'traffic sign'.

Figure 12: Sample of classes 'person' and 'car'.

Figure 13: Sample of class 'bus'.

Where the blue pixels refer to class ‘bus’, the red ones are the class ‘people’, yellow for ‘stop sign’, and class ‘car’ is represented by green.

3. SlowFast Networks

Each video of the dataset has an average duration of 7 seconds, and 64 frames are chosen from each clip that the model will process as images. The model, having been developed by Facebook's Artificial Intelligence team, is coded in PyTorch library instead of TensorFlow, which has been the library of the previous two models.

The base conditions have been the following:

- Optimizer: Stochastic Gradient Descent (SGD).
- Learning rate: 0.01
- Weight decay: 0.00005
- Batch size: 8

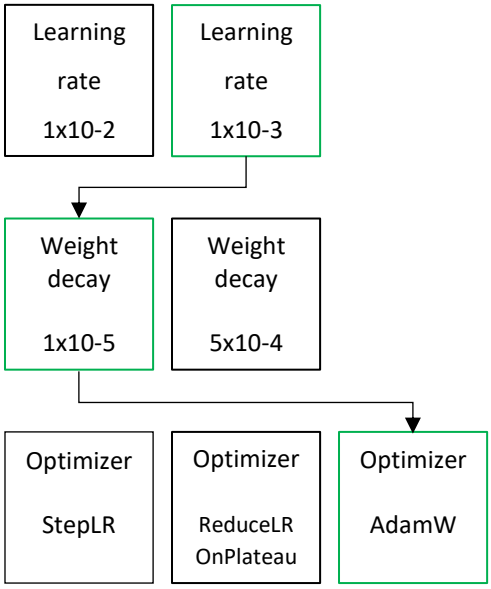

Figure 14: structure followed to optimize the model.

The optimizer was chosen to replicate the conditions from the paper. However, throughout the development of the project, more powerful optimizers have been analyzed depending on the application. For this reason, it was decided to analyze ReduceLROnPlateau and AdamW, being the latter the optimizer with the best results.

The final accuracy of the model has been 55%, compared to the 75% from the paper. Though, the parameters necessary

to minimize this difference are analyzed in the project.

In this case, the model classifies the content of the video, without adding boxes or classifying each pixel in the picture, as in the previous case. However, the possibility of adding an image classification model, such as Faster-RCNN, is analyzed during the project.

Integrating both models, the algorithm is able to process the content of the video and localize the objects in determined frames, as the next image shows:

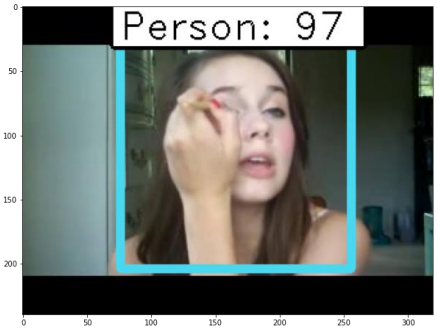

Figure 15: sample of class 'Apply Make Up'.

Figure 16: Sample of class 'Punch'.

IV. Conclusions

In this project, innovative computer vision algorithms have been analyzed. These algorithms introduce new techniques that will mark the development of this field in the coming years. Thus, it has been possible to better understand the operation of these algorithms, as well as the convolution neural networks, so often called 'black boxes' for their complex operation.

The results obtained have been compared with those in the state-of-the-art literature, analyzing the causes of the differences and understanding their operation. The great limitation when training has been the availability of resources, in this case of a GPU. The project includes an analysis of the different Cloud options if a personnel GPU is not possible, or to complement it, you do not have your own GPU or want to complement it with one, since very often the resources available in the state of the art are unaffordable for an average user, undermining the result.

Although Faster-RCNN got worse results than U-Net and needed more time to train, the integration of Faster-RCNN with video algorithms, as SlowFast in this project, proves that these models are powerful and scalable for several applications.

In the case of U-Net, the size of the model and its simpler complexity made possible to simulate multiple scenarios and test the effect of different hyperparameters. The main advantage of U-Net in comparison with similar algorithms in the state-of-the-art literature is the advantage of learning the context information associated with the image. This is done by the lateral connections between the down-sampling and the up-sampling paths. However, it has been observed that if the input image is too small this information may get lost. Although the applications are different, the same optimal image size has been obtained in this project and in the paper [2], proving that the results shown in the project are reliable.

Results obtained with SlowFast are positive in terms of computational resources and its application for action recognition. The model got an accuracy of almost 60% for the 101 classes of UCF-101 in just 6h. Therefore, training the model with AVA dataset, which includes bounding boxes for action detection, would allow the model to eb useful for more complex applications.

V. References

[1] S. Ren, K. He, R. Girshick y J. Sun, Faster R-CNN: Towards Real-Time Object, 2016.

[2] O. Ronneberger, P. Fischer y T. Brox, U-Net: Convolutional Networks for Biomedical, 2015.

[3] C. Feichtenhofer, H. Fan, J. Malik y K. He, SlowFast Networks for Video Recognition, 2019.