Vision and Perception
Project - Phase A

Karim Ghonim - Matricola: 1774086
Hossam Arafat - Matricola: 1803850
Ekin M. Senler- Matricola: 1801499
Cecilia A. Peredo - Matricola: 1822225
Huriye Kayis - Matricola: 1802952

27 June 2018

SAPIENZA
UNIVERSITÀ DI ROMA

1 Introduction

The goal of this project is to create a neural network capable of classifying between different dance activities; Tango, Breakdancing, Belly dancing, Ballet, Baton twirling, Cumbia and Cheerleading. This report covers Phase A of the project, namely instance segmentation of the key objects required to differentiate between activities which is necessary for Phase B, the actual identification of each dance. The network was trained on some objects from Coco dataset, as well as, the key objects previously mentioned, which was created manually on Labelbox. Different experiments were carried out in order to test the performance of the network.

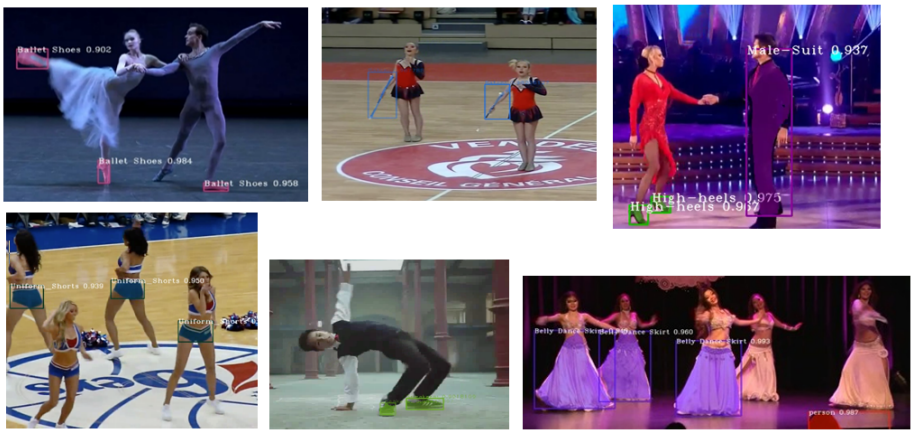


Figure 1: Network's Output

2 Related work

2.1 Transfer Learning

Transfer learning is related to problems such as multi-task learning and concept drift and is not exclusively an area of study for deep learning. Nevertheless, transfer learning is popular in deep learning given the enormous resources required to train deep learning models or the large and challenging datasets on which deep learning models are trained. Transfer learning only works in deep learning if the model features learned from the first task are general.
In practice, it is very hard to train entire CNN from scratch, since it is really rare to have database with sufficient size. There are multiple datasets with pretrained object detection networks like COCO, ImageNet, Oxford VGG etc. We took weights of pre-trained COCO dataset network and use it as our starting weights. By doing transfer learning, we save a lot of time and computational power. Modern ConvNets take 2-3 weeks to train across multiple GPU on ImageNet. On top of that actually, this approach allows us to take layer or multiple layers of the network and fine tune it for our usage. How do you decide what type of transfer learning you should perform on a new dataset? This is a function of several factors, but the two most important ones are the size of the dataset (small or big), and its similarity to the original dataset (e.g. ImageNet-like in terms of the content of images and the classes, or very different, such as microscope images).
Here are some common rules of thumb for navigating the 4 major scenarios:

1. New dataset is small and similar to original dataset. Since the data is small, it is not a good idea to fine-tune the COCO due to overfitting concerns. Since the data is similar to the original data, we expect higher-level features in the COCO to be relevant to this dataset as well. Hence, the best idea might be to train a linear classifier on the CNN codes.

2. New dataset is large and similar to the original dataset. Since we have more data, we can have more confidence that we won't overfit if we were to try to fine-tune through the full network.

3. New dataset is small but very different from the original dataset. Since the data is small, it is likely best to only train a linear classifier. Since the dataset is very different, it might not be best to train the classifier form the top of the network, which contains more dataset-specific features. Instead, it might work better to train the SVM classifier from activations somewhere earlier in the network.

4. New dataset is large and very different from the original dataset. Since the dataset is very large, we may expect that we can afford to train a COCO from scratch. However, in practice it is very often still beneficial to initialize with weights from a pretrained model. In this case, we would have enough data and confidence to fine-tune through the entire network.

In our case, information on the coco dataset not be beneficial to the current task. Surely, in the model there are weights that distinguish between cats and dogs. This is probably irrelevant to dance type classification. To further better our network, we unfreeze the last few layer of the R-CNN and retain them. (CNN features are more generic in early layers and more original-dataset specific in later layers).

2.2 Mask R-CNN

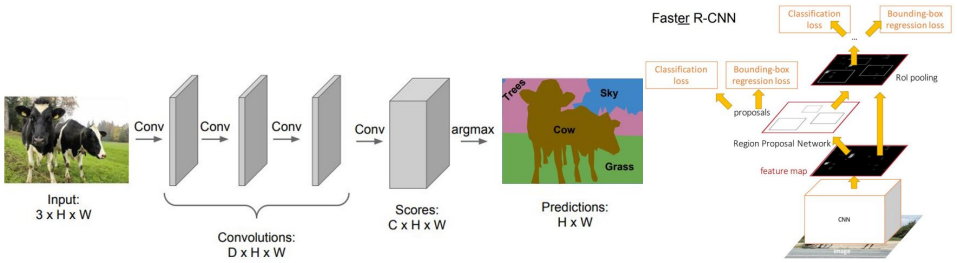
Mask R-CNN is a type of Region-Based Convolutional Neural Network (R-CNN) developed by Facebook Artificial Intelligence Research (FAIR) to tackle the challenging problem of Instance segmentation. Instance segmentation is the task of localizing instances and identifying instance outlines at the pixel level inside the image. Compared to similar computer vision tasks, it's one of the hardest but it can be divided into 2 tasks: Semantic Segmentation of every pixel in the image whilst differentiating between instances, and Object Detection (Classification and Localization of a variable number of instances per image).

Semantic Segmentation can be handled by Fully Convolution Networks (FCN), as the task warrants preserving spatial features of the input as well as the pixel-to-pixel alignment. The input image is fed to the network where the final convolution layer of the network outputs a [C x H x W] tensor where C is the number of categories that we care about. This tensor can be simply viewed as giving classification scores for every pixel in the input image at every location in the input image.

Meanwhile, Object detection can be handled very efficiently by Faster Region-based Neural Networks. These types of CNNs feed the entire input image altogether through some convolutional layers to get a convolutional feature map representing the entire high resolution image. This is then passed to a Region Proposal Network (RPN) that is trained to look for interesting features and propose "Regions of Interest" in the image where objects are most likely to be found.

All regions proposed by the RPN are wrapped into a fixed size using "ROI Pooling" and fed into the final Fully connected layers which use a Support Vector Machine to classify the object instance inside each region and regresses over 4 numbers which represent the offsets / corrections needed to transform the center coordinates, width, and height of the proposed region bounding boxes into the final bounding box around objects.

Consequently, Mask R-CNN has a ResNet CNN for its base architecture followed by 2 branches that combine "the best of both worlds" by using the best network architectures used for Object Detection and Semantic Segmentation, namely, Faster R-CNN with "ROIAlign" in place of "ROI Pooling" to preserve pixel-to-pixel Alignment and FCNs.



(a) Semantic Segmentation using a simple Fully Convolution Network (FCN).

(b) Faster R-CNN utilizes a Region Proposal Network (RPN) that is trained to identify Regions of interest (ROIs).

Figure 2: The two branches Mask R-CNN is built upon.
(a) FCN (b) Faster R-CNN

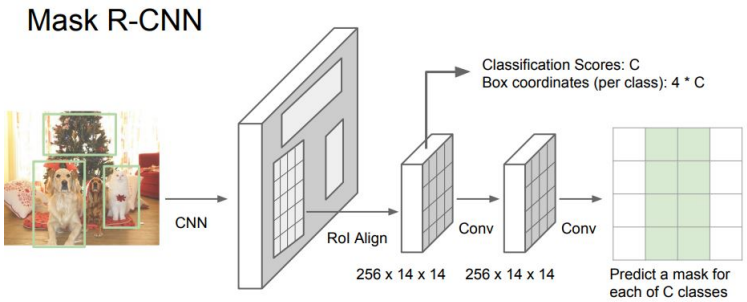


Figure 3: Mask R-CNN architecture is built upon Faster R-CNN to classify and localize multiple object instances per image in addition to a Fully Convolution Network to preform Semantic Segmentation.

3 Dataset

3.1 Coco dataset

COCO is a large dataset created by Microsoft in order to provide the computer vision community with a reliable dataset that can be used to tackle the problem of instance segmentation. Other large datasets exist such as ImageNet, PASCAL or SUN. MS COCO was chosen for several reasons; the vast collection of object instances it provides for everyday life images in their natural environments and with varied viewpoints (in the background, partially occluded, among clutter), the rigorous process followed in creating the dataset, from training the workers responsible for segmenting the instances, to utilizing over 70,000 worker hours while maintaining a high level of attention to detail, supervision and quality control.

3.2 Dance dataset - Labelbox

Labelbox was used for the creation of Dance dataset. It provides a simple interface to create and manage training data for expert AI systems. First, images were uploaded to Labelbox and then each occurring instance was segmented. The output was in the form of a Json file containing all required information about the images, masks, and labels. Each dance activity was studied and objects unique to each dance were selected accordingly, per Table 1. It is to be noted that after thorough consideration no key objects were selected for Cumbia, therefore, it will be handled in Phase B.

Cheerleading	White Sport Shoes	Shorts	Uniform Skirts
Tango	Male Suite	High Heels	-
Breakdancing	Sneakers	Hats	-
Belly dancing	Belly Dancing Skirt	Barefoot	
Ballet	Ballet Shoes	-	-
Baton Twirling	Baton	-	-
Cumbia	-	-	-

Table 1: The objects specified for each domain

The guidelines introduced by Coco were followed in the creation process of the dataset.

4 Implementation

4.1 Dataset Preparation

The dataset used for training the network is comprised of 2367 images, 35% Coco (2014) and 65% Dance images. Only 21 classes were picked from Coco dataset; [Background, person, bicycle, car, motorcycle, bus, traffic light, fire hydrant, stop sign, parking meter, bench, backpack, handbag, tie, sports ball, baseball bat, skateboard, bottle, potted plant, tv, and clock], with 42 images for each class, totaling 840 images. All images chosen contained 3 masks or less as some images for Coco contains too many masks, e.g. images of crowds or banana trucks. As for Dance dataset, 1,697 images were used containing a total of 3,517 instances or masks. From the Json file exported for Dance dataset, all the images and their corresponding masks were generated and then split into 90% training and 10% validation.

4.2 Training

Talk about everything from hyper parameters, epoches etc. to classes used
The weights of the network pre-trained on the MS CoCo dataset and Transfer learning was used to fine-tune it on the Dance dataset. - excluding the weights for layers mentioned
An NVIDIA Tesla GPU with 6GB of Memory was used to train the network, consequently, it was only possible to load 1 image per time step.
1,500 steps per epoch were used, any detections with confidence less than 90% were ignored, and the learning rate was set to 0.001. In order to improve performance, the dataset was augmented by flipping the images left and right 50% of the time. The network's pre-trained weights on Coco dataset were used and fine tuned in order to include the new objects from Dance dataset. As the first step of fine tuning, the weights for (mrcnn-class-logits, mrcnn-bbox-fc, mrcnn-bbox, mrcnn-mask) layers were excluded. The training process is comprised of three stages, training the network heads, then fine-tuning from ResNet stage 4 and finally, all layers are fine-tuned. The first two stages use the aforementioned learning rate, while the third stage divides it by 10.

4.3 Experiments

No. of Epochs		
	Experiment 1	Experiment 2
Stage 1	30	35
Stage 2	30	40
Stage 3	10	15

Table 2: Number of epochs for each training experiment

Two experiments were conducted, during the first experiment, the first and second stage lasted for 30 epochs while the last lasted 10. In the second experiment, the three training stages lasted 35, 40, and 15 epochs respectively, as shown in Table 2. All other hyper-parameters used kept constant for both experiments, in order to test the improvement acquired by fine-tuning for a longer time.

4.4 Evaluation

The evaluation of the network's performance was performed through both, instance segmentation of an actual video, and the accuracy of segmenting and classifying the instances found in the validation set. The validation set was used in place of the evaluation set in order to provide as many images as possible for training the network. This is also because the losses calculated for the validation while training were erratic, inaccurate and unrepresentative of the network's performance due to the fact that the network detects Coco objects in the validation set, that are missing from the ground truth, as only key objects for dancing activities were segmented and labeled.

After careful review and testing on many videos with different environments, quality and lighting, the network generally showed stable and consistent performance in detecting and segmenting the objects, whether from Coco or Dance datasets. However, some reoccurring issues were encountered, and will be discussed in the following section.

As for the evaluation, weights obtained from both experiments were tested in order to observe the effect of training the network, and fine-tuning the weights for more epochs. The only hyper-parameter changed through the evaluation experiments was minimum detection confidence level, the results are provided in Table 3.

Opposing to matterport's evaluation of their network which was based on the mean average precision (mAP), in order to avoid the problem mentioned earlier regarding the unsegmented Coco objects in the validation set, the accuracy was measured based on three criteria; the area of the segmented object is to be between 50% and 150% of the actual area of the object in the ground truth, the centroid detected by the network falls inside the bounding box of the mask in the original image, and finally, the class assigned by the network is correct. If all three criteria are met, then the instance segmentation performed by the network is deemed successful.

	Accuracy (%)	
Confidence Level (%)	Experiment 1	Experiment 2
60	69	TBA
70	65	TBA
90	52	TBA

Table 3: Number of epochs for each training experiment

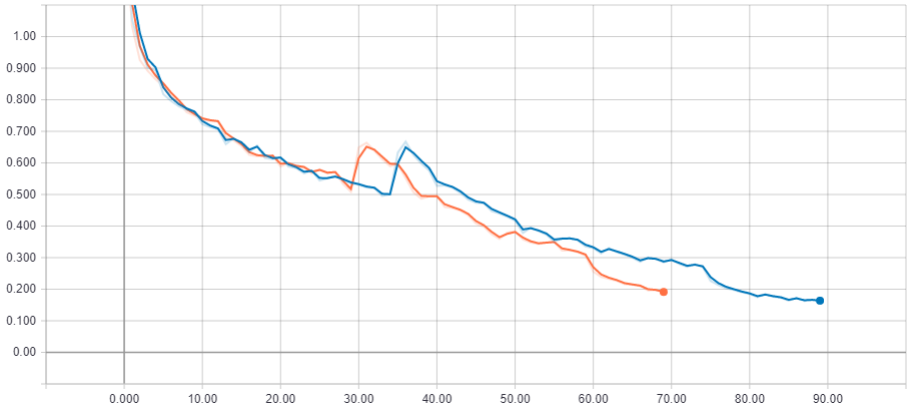


Figure 4: Graph of the Loss evolution during both training experiments. Note the peaks that occur when transitioning from one training stage to the next.

4.5 Problems faced - Video segmentation

After evaluating and analysing multiple videos for each category domain, we encountered encouraging results as well as some errors. Generally, the network performed very well throughout the videos to detect objects. When the videos were in a clear setting and recorded in high quality, the output was

more precise and able to encounter the objects easily. The images shown in Figure [3] demonstrate how well our system detects objects for each domain.

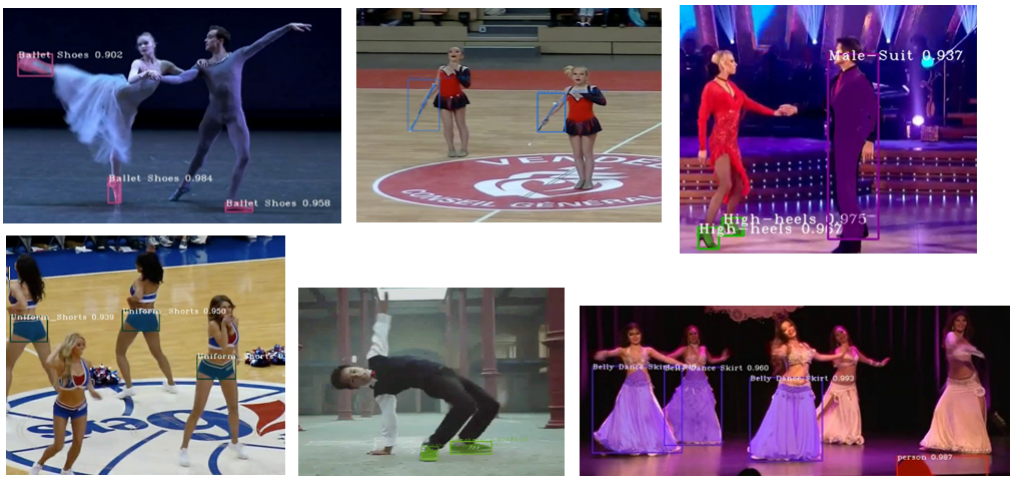


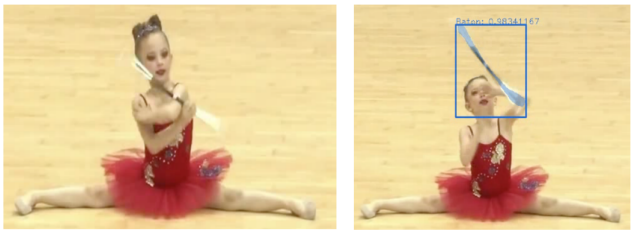
Figure 5: Images from each domain video showing the correctly identified objects. From the left top image to the bottom left, the domain and objects identified are the following: "Ballet" - ballet shoes, "Baton Twirling" - baton, "Cheerleading" - shorts, "Tango" - high heels and male suit, "Breakdancing" - sneakers, and "Belly dancing" - skirt.

Some issues that we identified during our evaluation were the quality and lighting of the videos, COCO dataset dominating our dataset, sneaker objects dominating across all shoes, long, and slim pole-looking objects being identified as batons. The following sections we will be discussing each of these issues separately.

4.5.1 Quality and Lighting of the video

After analysing the detection of the objects in the videos, it has been observed that using high-quality videos and having good lighting are the most important aspects needed for the network to have a higher performance at detecting objects. Since the network looks at the videos frame by frame, the quality of each frame and therefore the visibility of the objects in them need to be clear in order to identify the objects. Otherwise, this causes the network to identify the objects as intermittently rather than steadily which is preferred.
A clear example of this occurring is shown in Figure 4(a), where the baton is initially not identified and then the next consecutive frame the baton is correctly identified. As seen in the image where the baton is not identified,

one can notice that the baton looks blurry due to its movement and therefore the network cannot identify it as a baton. During the labelling of masks, we considered these movements and included them in our set. However, with lower frames per second (fps) in videos there is a significant distortion of these objects which makes detection harder or even impossible in certain cases.
Also making detection harder is the lighting in the videos. As seen in Figure 4(b), poor lighting in videos do not enable objects to be identified. In this case, the ballet shoes are not identified since the shoe's accurate outline and details are not visible in such low light setting. Another problem is a setting where there is too much brightness, since the contrast between, for example, the shoes and the floor or their uniform are not evident.



(a) Images from a video showing two consecutive frames where in one there is no detection of objects due to blurriness and the next frame where a baton is correctly detected.

(b) An image from a video showing no detection in a low light setting.

Figure 6: Effects of Quality and Lighting

4.5.2 CoCo dataset dominating our objects

The second issue encountered during the analysis of the videos is that the objects labelled by the Coco dataset dominates over objects from Dance dataset. As seen in Figure 5, the object in the scene is a baton but it is identified as sports ball because of the rounded ends and baseball bat because of the long stick. Even though Coco images take up only 35% from the whole dataset, the fact that the weights originally used were extremely well trained on Coco dataset, that the network learned to find those objects much easier and with much higher confidence that it is sometimes misleading.

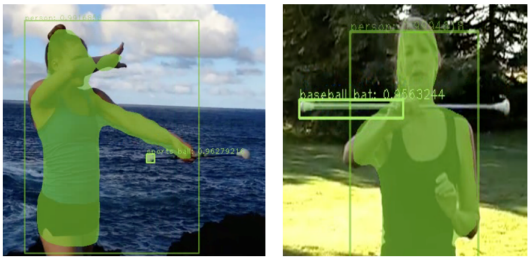


Figure 7: Images from a video showing the baton being misidentified as objects from the CoCo dataset.

4.5.3 Sneakers dominating within our dataset

Similar to the COCO dataset dominating over our objects, sneakers also dominate within our dancing dataset when locating shoes or feet. The number of instances of sneakers in the dataset is higher in comparison to that of ballet shoes, high heels, and bare feet, simply for the nature of the activity, that is usually in big groups or teams sometimes. This quantity gap between the different types of shoes used for the training is the reason the network confuses the objects at times, and has a bias towards detecting sneakers more often than it detects the other shoe types.
As seen in Figure 6, sneakers are detected incorrectly in other dancing domains like belly dancing, ballet, and tango. For instance, for the middle image the ballet shoe should have been identified but instead it identified it as a sneaker for some frames. This is due to both the bias towards sneakers and its location in the background where the detail of the shoe is lost, which makes it harder for the network to output an accurate classification of the object detected.

Figure 8: Images from different videos showing the misidentification of sneakers instead of barefoot, high heels and ballet shoes, respectively.

4.5.4 Long slim objects detected as "Baton"

The fourth and last issue observed in our videos was the challenging task to detect batons for many items that resembled its shape as a tiny, straight pole since many items can be confused as this. Even though the network is successful to detect baton in its domain, it is a conflicting issue for other domains as in any kind of scene it is easy to find tiny straight objects. We give some instances of this issue from different perspectives in the following examples.

Two examples of this issue are seen in Figure 7. As for the right image and in many other videos, the network detects uniform stripes as batons. The cheerleader's uniform is detected as a baton since cheerleading uniforms mainly contain white stripes on it. Similar to this, the network detects arms as batons when it has one color throughout the arm and is positioned straight. For the image on the left, a person is seen holding the bar, but the network detects a baton. This is expected as the bar has a similar geometric shape with a baton in a 2D perspective. This means that many things could be interpreted as a baton as it is only a straight pole.

Figure 9: Images showing tiny straight objects detected as a baton. The image on the left shows the bar in a ballet scene misidentified as a baton. On the right image, a cheerleader's arm is detected as a baton.

5 Future work and improvements

With Phase A complete, the network is able to perform image segmentation for all required objects successfully, therefore, the prerequisite for Phase B is fulfilled, which is going to be the actual activity classification performed through the implementation of an LSTM.

Due to a lack of computing power, it wasn't possible to perform more experiments, in order to find the optimum hyper-parameters. The network could have been also trained for a longer period. Improvements could also be made for the dataset, as only the key objects required for dancing were segmented and labeled by Labelbox, therefore, ignoring objects that were included in Coco which will cause a bit of confusion for the network, as it will detect a person but will not find that instance in the ground truth of the image. With more working hours and a training course for the team, the segmentation would have been much more even through out all the objects, and would have provided a larger dataset, which would have increased performance

6 Conclusion

The proposed solutions and techniques for the required improvements to the program were reached after several experiments, as they simply provided the most stable performance and were more robust to the changes that could be implemented for future improvement of the program.