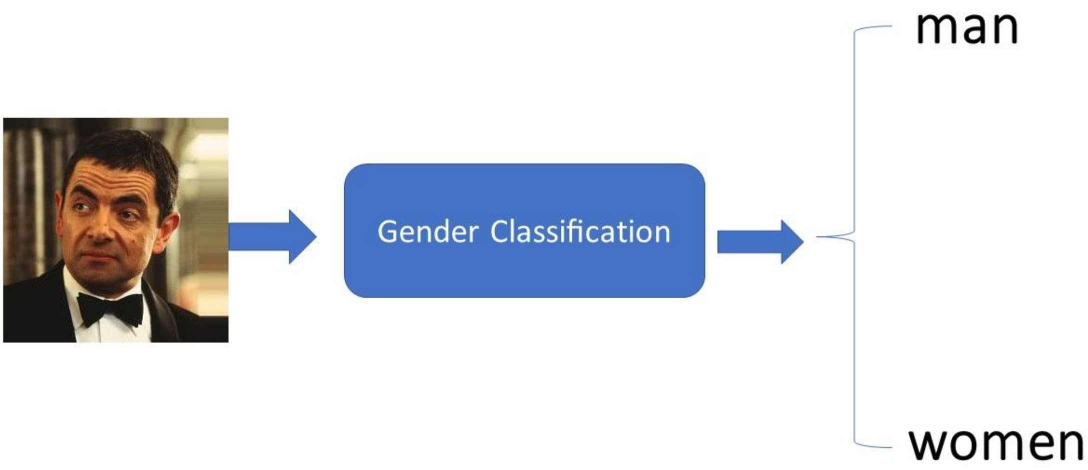
Convolutional Neural Networks for Gender Classification

1 Introduction:

The Recognition technology is related to many areas of public life, image recognition technology is an important branch of computer vision, neural network image recognition technology is along with the modern computer technology, image processing and artificial intelligence. The success of facial recognition feature relies on a seamless robot for user interaction. So that the robot can extract information from the faces in the images, for example identify gender in pictures. The difficulty of detecting gender has been demonstrated by the use of machine learning technology within each task{1}. This leads to models with millions of parameters trained under thousands of samples {2}. People can imagine the difficulty of framing a picture on two types of man or woman with thousands of the pictures. But despite challenges and difficulties, robots are able to perform face recognition functions by means of advanced recognition systems, capable of computational strength and power computability. Also, all the modern styles that are related to the images are using the technique the Convolutional Neural Networks (CNNs). These tasks require CNN architectures with millions of parameters, therefore their deployment in robot platforms and real-time systems.



Convolutional Neural Networks are expressions of deep neural networks and have an important classification of images (e.g. animal identification) and select elements within images and perform object recognition within scenes. They are algorithms that can identify faces, individuals, street signs, tumors, platypuses and many other aspects of visual data. Convolutional networks perform optical character recognition (OCR) to digitize text and make natural-language processing possible on analog and handwritten documents, where the images are symbols to be transcribed.

CNNs can also be applied to sound when it is represented visually as a spectrogram. More recently, convolutional networks have been applied directly to text analytics as well as graph data with graph convolutional networks. The efficacy of convolutional nets (ConvNets or CNNs) in image recognition is one of the main reasons why the world has woken up to the efficacy of deep learning. They are powering major advances in computer vision (CV), which has obvious applications for self-driving cars, robotics, drones, security, medical diagnoses, and treatments for the visually impaired.

2 Datasets and Inputs:

In this project we have used the publicly available image dataset for gender classification from (IMDB), This file contains 7GB pictures of artists and celebrities[here] . The file contains only faces with a file metadata , also provide a version with the cropped faces (with 40% margin). This version is much smaller.

mat file which can be loaded with Matlab containing all the meta information. The format is as follows:

- **dob:** date of birth (Matlab serial date number)
- **photo_taken:** year when the photo was taken
- **full_path:** path to file
- **gender:** 0 for female and 1 for male, *NaN* if unknown
- **name:** name of the celebrity
- **face_location:** location of the face. To crop the face in Matlab run
- **face_score:** detector score (the higher the better). *Inf* implies that no face was found in the image and the *face_location* then just returns the entire image
- **second_face_score:** detector score of the face with the second highest score. This is useful to ignore images with more than one face. *second_face_score* is *NaN* if no second face was detected.
- **celeb_names (IMDB only):** list of all celebrity names
- **celeb_id (IMDB only):** index of celebrity name

3 Solution Statement:

As I mentioned previously, I would like to use deep learning as a final solution. The reason is that deep learning is enriched and flexible in identifying the special features in the image and all the aspects that can solve the problem of identifying gender. I plan to use a convolutional neural network (CNN), which are very effective at finding patterns within images by using filters to find specific pixel groupings that are important. My aim is to be both: more effective at detecting the lines, as well as faster at doing so than common computer vision techniques.

4 Evaluation Metrics:

The common use of CNN is to extract the feature that includes a set of layers that are linked together at the end. Fully connected layers tend to contain most of the parameters in a CNN, especially VGG16 [4]. The last connected strata contains about 90% of the total parameters. Modern methods have reduced the number of parameters in the end such as Inception V3 [5] by including a Global Average Pooling operation. Global Average Pooling reduces each feature map into a scalar value by taking the average overall elements in the feature map. The average operation forces the network to extract global features from the input image.

5 Project Design:

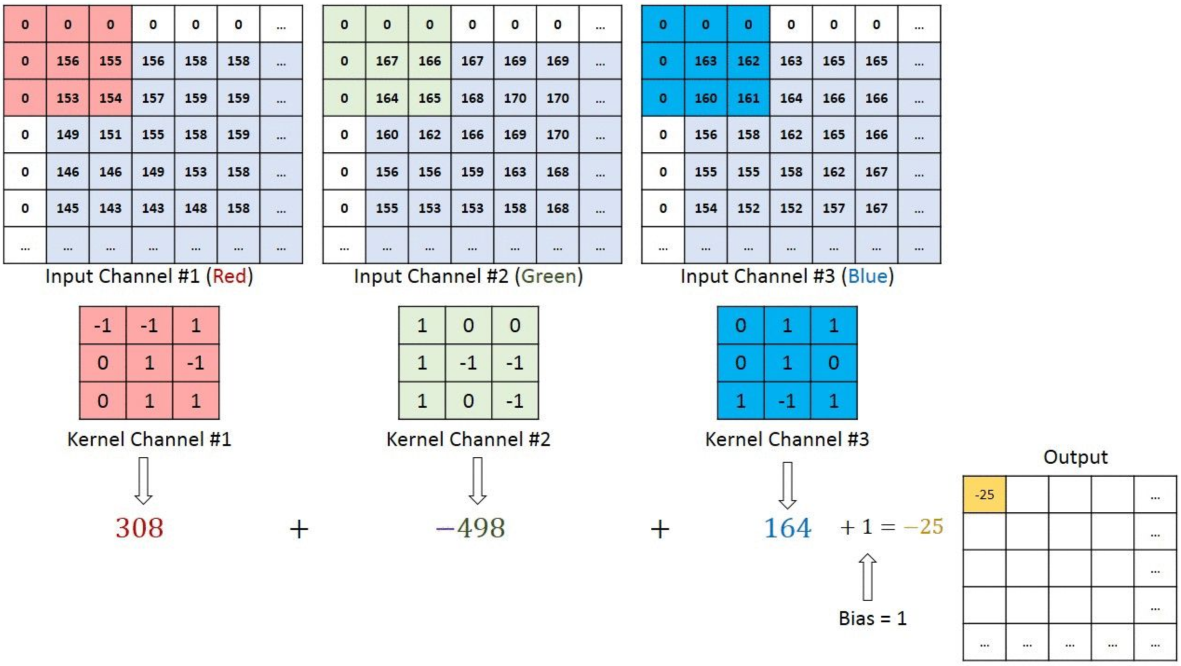
The first thing to know about convolutional networks is that they don’t perceive images like humans do. Therefore, you are going to have to think in a different way about what an image means as it is fed to and processed by a convolutional network.

Convolutional networks perceive images as volumes, i.e. three-dimensional objects, rather than flat canvases to be measured only by width and height. That’s because digital color images have a red-blue-green (RGB) encoding, mixing those three colors to produce the color spectrum humans perceive. A convolutional network ingests such images as three separate strata of color stacked one on top of the others. So a convolutional network receives a normal color image as a rectangular box whose width and height are measured by the number of pixels along those dimensions, and whose depth is three layers deep, one for each letter in RGB. Those depth layers are referred as *channels*.

As images move through a convolutional network, we will describe them in terms of input and output volumes, expressing them mathematically as matrices of multiple dimensions in this form: 30x30x3. From layer to layer, their dimensions change for reasons that will be explained below. You will need to pay close attention to the precise measures of each dimension of the image volume, because they are the foundation of the linear algebra operations used to process images.

Now, for each pixel of an image, the intensity of R, G and B will be expressed by a number, and that number will be an element in one of the three, stacked two-dimensional matrices, which together form the image volume. Those numbers are the initial, raw sensory features being fed into the convolutional network and the ConvNets purpose is to find which of those numbers are significant signals that actually help classify images more accurately (just like other feedforward networks we have discussed).

Rather than focus on one pixel at a time, a convolutional net takes square patches of pixels and passes them through a *filter*. That filter is also a square matrix smaller than the image itself, and equal in size to the patch. It is also called a *kernel*, which will ring a bell for those familiar with support-vector machines, and the job of the filter is to find patterns in the pixels.



0	0	0	0	0	0	...
0	156	155	156	158	158	...
0	153	154	157	159	159	...
0	149	151	155	158	159	...
0	146	146	149	153	158	...
0	145	143	143	148	158	...
...	...	...	...	...	...	...

Input Channel #1 (Red)

0	0	0	0	0	0	...
0	167	166	167	169	169	...
0	164	165	168	170	170	...
0	160	162	166	169	170	...
0	156	156	159	163	168	...
0	155	153	153	158	168	...
...	...	...	...	...	...	...

Input Channel #2 (Green)

0	0	0	0	0	0	...
0	163	162	163	165	165	...
0	160	161	164	166	166	...
0	156	158	162	165	166	...
0	155	155	158	162	167	...
0	154	152	152	157	167	...
...	...	...	...	...	...	...

Input Channel #3 (Blue)

-1	-1	1
0	1	-1
0	1	1

Kernel Channel #1

1	0	0
1	-1	-1
1	0	-1

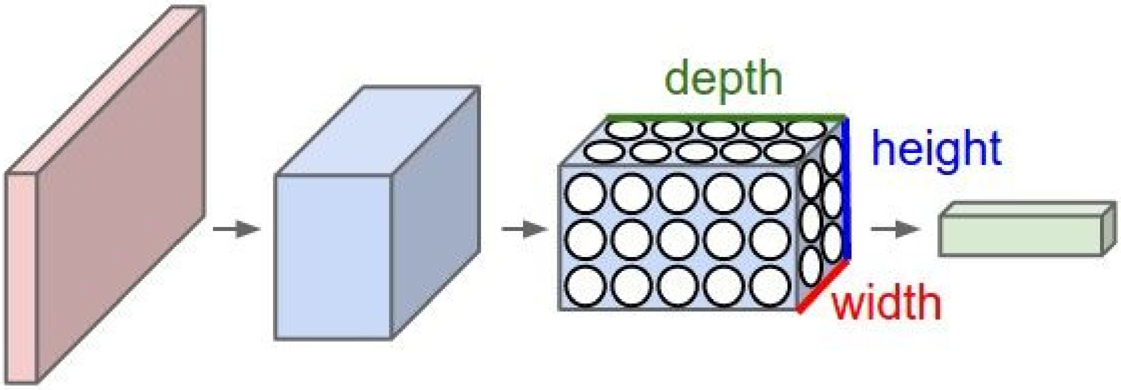
Kernel Channel #2

0	1	1
0	1	0
1	-1	1

Kernel Channel #3

Output

-25				...
				...
				...
				...
...	...	...	...	...



I have set the form on depending on the test frequency and the number of parameters, I have built the model based on the number of parameters as well as the accuracy of these parameters in the model. First, the use of small CNNs alleviate slow performances in hardware-constrained systems such robot platforms. Second, the reduction of parameters provides a better generalization under an Occam’s razor framework, Our first model reliesontheideaofeliminatingcompletelythefullyconnected layers[8]. The second step is to train each class with an optimizer [ADMS] and then remove the layers connected with each other we are using. Global Average Pooling was achieved by having in the last convolutional layer the same number of feature maps as number of classes, and applying a softmax activation function to each reduced feature map. Our initial proposed architecture is a standard fully-convolutional neural network composed of 9 convolution layers: ReLUs [6], Batch Normalization [7] and Global Average Pooling. This model contains approximately 600,000 parameters. It was trained on the IMDB gender dataset, which contains 460,723 RGB images where each image belongs to the class “woman” or “man”, and it achieved an accuracy of 96% in this dataset.

Our second model is inspired by the Xception [8] architecture. This architecture combines the use of residual modules [10] and depth-wise separable convolutions [9]. Residual modules modify the desired mapping between two subsequent layers, so that the learned features become the difference of the original feature map and the desired features. Consequently, the desired features H(x) are modified in order to solve an easier learning problem F(X) such that:

$$H(x) = F(x) + x$$

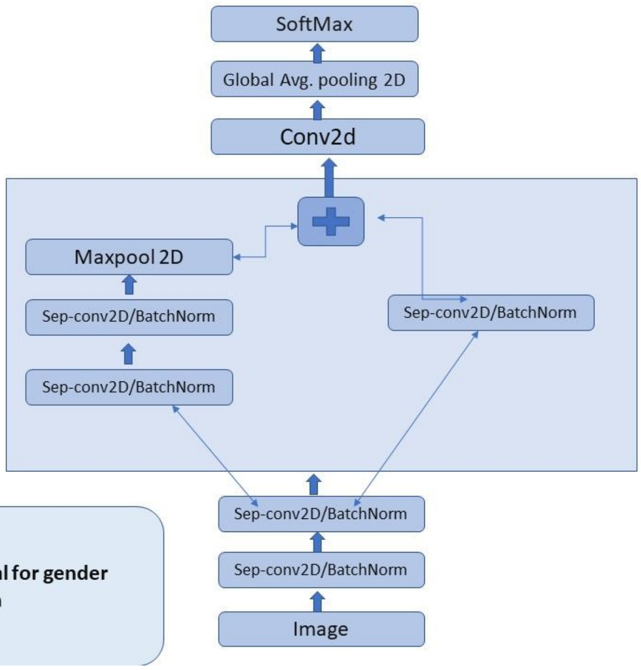


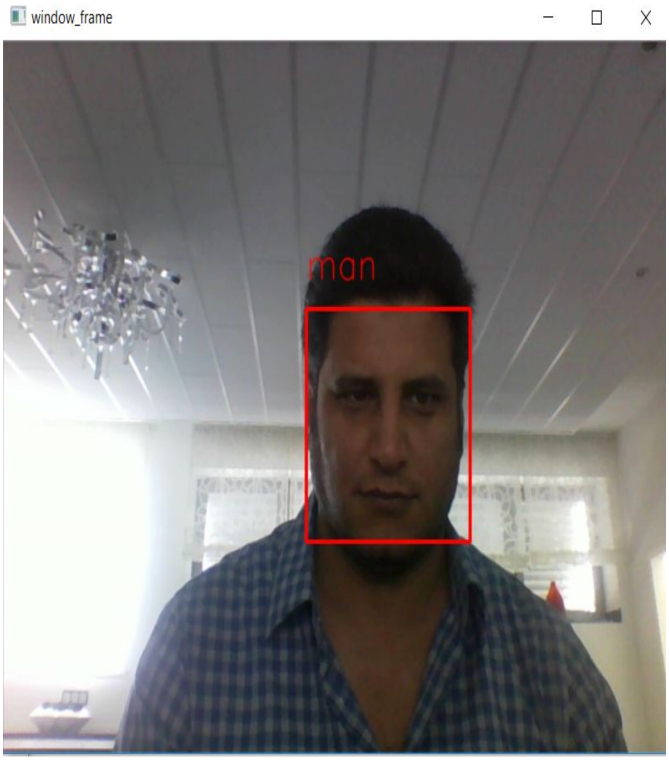
Figure 3: Model proposal for gender classification

Our final architecture is a fully-convolutional neural network that contains 4 residual depth-wise separable convolutions where each convolution is followed by a batch normalization operation and a ReLU activation function. The last layer applies a global average pooling and a soft-max activation function to produce a prediction. This architecture has approximately 60000 parameters, which corresponds to a reduction of 10× when it is compared to our initial naive implementation,and 80×

when it is compared to the original CNN. Figure 3 displays our complete final architecture which we refer as mini-Xception. This architectures obtains an accuracy of 95% in gender classification task. Which corresponds to a reduction of one percent with respect to our initial implementation.

6 RESULTS

We have obtained positive results through the computer camera(10) if it is possible to separate the genders (man and woman). It can be observed that Network (CNN) was able to identify certain characteristics in the face of the human through which we have been able to sort out the gender. Taking into account that ordinary man finds it difficult sometimes to differentiate between men and women because of the overlap in the features of the beard, hair, age, etc., the network (CNN) finds it difficult to differentiate between gender.



REFERENCES

[1] Ian Goodfellow et al. Challenges in Representation Learning: A report on three machine learning contests, 2013.[2] Andrew G. Howard et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861, 2017

[3] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2818–2826, 2016

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[5] Dario Amodei et al. Deep speech 2: End-to-end speech recognition in english and mandarin. CoRR, abs/1512.02595, 2015.

[6] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pages 315–323, 2011.

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning, pages 448–456, 2015.

[8] Franc¸ois Chollet. Xception: Deep learning with depthwise separable convolutions. CoRR, abs/1610.02357, 2016

[9] Andrew G. Howard et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861, 2017.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.