Interpretation and Evaluation of Image level Human gender and emotion detection

Tapan Taranath Hegde
HOCHSCHULE RAVENSBURG-WEINGARTEN
Weingarten, Germany
tapanhegde93@gmail.com

***Abstract*—When interfaces are responsive to human emotion or stress, human-computer interaction can feel normal and successful. Emotion recognition models are increasingly being used by intelligent systems to enhance their interactions with humans. This is significant because systems can adjust their responses and behavioral behaviors in response to human emotions, making interactions more natural.Emotions can be expressed by speech, body movements, and facial expressions. As a consequence, the relationship between human and computer communication places a focus on extracting and interpreting emotion. The same can be said about recognizing a person's gender; if the gender of the human is recognized, the computer can still propose better alternatives or have a better interaction.**

The project's aim is to compare and test two separate models of human gender and emotion detection at the image level. Model 1 uses the FaceNet model to detect facial coordinates and the MXNet facial attribute extraction model to extract 40 different facial attributes. Emotion, Age, and Gender, as well as facial characteristics, are all detected by this solution. Model 2 employs 3x3 convolutional layers stacked in rising depth on top of each other. Max pooling is used to reduce the size of a volume. The softmax classifier is then followed by two completely connected layers, each with 4,096 nodes. The Resnet architecture is then used to detect gender and emotion.

***Index Terms*—Gender, Emotion, FaceNet, MXNet, softmax, Resnet**

I. Model 1

This model basically works in 3 steps; Face Detection using FaceNet Model then it Predicts gender of the face and subsequently detects the emotion of the face. Deep Learning Models used for the library are; FaceNet model used for facial landmark recognition, a trained lightened moon Mxnet model used for facial attribute extraction, real-time face detection and emotion/gender classification using fer2013/IMDB datasets with a keras CNN model and openCV, AgeNet pre-trained caffe model used for Age detection, GenderNet caffe model used for Gender detection.

A. FaceNet model

1) Introduction : This method uses a deep convolutional network to learn a Euclidean embedding for each image. Face similarity is directly correlated by the squared L2 distances in the embedding space: faces of the same individual have small distances, whereas faces of different people have wide distances. If this embedding is developed, the aforementioned tasks become simple: face verification is as simple as thresholding the distance between the two embeddings; recognition is reduced to a k-NN classification problem; and clustering can be accomplished using off-the-shelf techniques like k-means or agglomerative clustering.

FaceNet uses a triplet-based loss function based on LMNN [1] to explicitly train its output to be a compact 128-D embedding. The loss attempts to distinguish the positive pair from the negative pair by a gap range, with the triplets consisting of two matching face thumbnails and one non-matching face thumbnail. Other than scale and translation, the thumbnails are tight crops of the face region with no 2D or 3D alignment.

2) Method : FaceNet is a machine learning algorithm that explicitly learns a mapping from face images to a compact Euclidean space in which distances are directly proportional to a measure of face similarity. Tasks like face recognition, verification, and clustering can be easily implemented using standard techniques with FaceNet embeddings as feature vectors once this space has been developed. A deep convolutional network is used by FaceNet. We aim for a f (x) embedding from an image x into a feature space R d such that the squared distance between all faces of the same identity, regardless of imaging conditions, is small, while the squared distance between a pair of face images from different identities is high. Despite the fact that it is not specifically comparable to other losses, such as those centered on pairs of positives and negatives, it is commonly presumed that the triplet loss is best suited to face verification. The theory is that losing allows all faces of a single identity to be projected onto a single point in embedding space. The triplet loss, on the other hand, attempts to create a margin between of pair of faces from one individual to all others. This allows one identity's faces to exist on a manifold while also enforcing the gap and discriminating against other identities.

3) Triplet Selection : Rather than choosing the toughest positive, all anchor positive pairs are used in a mini-batch while the hard negatives are still selected. When comparing hard anchor-positive pairs to all anchor-positive pairs within a mini-batch, it was discovered that the anchor positive form is more robust in practice and converges slightly faster at the start of training. Selecting the most difficult negatives will result in poor local minima early in training, specifically a collapsed model (i.e. f (x) = 0). To help reduce this, choose x ni in such a way that

$$||f(x_i^a) - f(x_i^p)||_2^2 < ||f(x_i^a) - f(x_i^n)||_2^2$$

These negative exemplars are semi-hard since they are further from the anchor than the positive exemplar, but they are also difficult because the squared distance is similar to the positive distance of the anchor. Those negatives are included inside the margin. As previously stated, proper triplet selection is critical for rapid convergence.Small mini-batches, which have been shown to increase convergence during Stochastic Gradient Descent (SGD), or batches of tens to hundreds of exemplars, could be more efficient[2]. The way we pick hard appropriate triplets from within the mini-batches, however, is the key restriction in terms of batch size. A batch size of about 1,800 exemplars is used in most experiments.

4) Deep Convolutional Networks : For face verification, the method requires learning an embedding into a Euclidean space directly. This distinguishes it from other methods [4] that rely on the CNN bottleneck layer or involve additional post-processing including multiple model concatenation and PCA, as well as SVM classification. This end-to-end training simplifies the setup while also showing that optimizing a loss that is important to the task at hand improves performance. Another advantage of our model is that it only necessitates a small amount of alignment (tight crop around the face area). For instance, [5] performs a complex 3D alignment.

5) Summary : This is a very useful model to employ. Small networks that can run on a cell phone and are compatible with a larger server-side model will also be interesting to train.

B. MOON Mxnet model

On the CelebA dataset, the MOON architecture achieves an accurate, computationally effective, and compact representation that advances the state of the art. In CelebA, dataset bias was explored, and domain adaptation methods for training to a different target distribution without using training samples from that population were proposed. Evaluations were performed on a novel re-balanced version of CelebA (the CelebAB dataset) and the LFW data set, combining domain adaptive methods and multiple-task goals into one blended objective feature.

By specifically forcing hidden layers in the network to integrate information from multiple labels while simultaneously imposing defined balance constraints through a domain adaptive loss, the approach implicitly leverages attribute correlations. Although CelebA labels are binary, MOON's weighted Euclidean loss allows labels to be learned over a continuous spectrum, which may be a better representation for certain attributes (e.g., Big Nose,Young). Experiments show that MOON not only improves the state of the art in facial attribute recognition, but it also outperforms DCNNs that have been trained independently using the same results.

C. CNN for Emotion and Gender Classification

In hardware-constrained systems, such as robot platforms, standard small CNNs relieve us of slow performance. Under an Occam's razor paradigm, the reduction of parameters allows for greater generalization. This model is based on the concept of removing all fully linked layers. The ADAM optimizer [8] was used to train the architecture. The final convolutional layer has the same number of feature maps as groups, and each reduced feature map is activated with a softmax activation function. There are approximately 600,000 parameters in this model. It was trained on the IMDB gender data set, which contains 460,723 RGB images classified as either "woman" or "male," and it achieved a 96 percent accuracy rate in this data.The FER-2013 data set was also used to test this model. This data set contains 35,887 gray scale photos, each of which is classified as "angry," "disgust," "fear," "happy," "sad," "surprise," or "neutral."

The design removed the last completely connected layer, reducing the number of parameters even further by removing them from the convolutional layers. This was achieved by using depth-wise separable convolutions. Convolutions that are depth-wise separable are made up of two layers: depth-wise convolutions and point-wise convolutions. The primary aim of these layers is to differentiate between spatial and channel cross correlations. As compared to regular convolutions, depth-wise separable convolutions minimize computation by a factor of N 1 +1 D 2 [2]. It is possible to see the difference between a regular Convolution layer and a depth-wise separable convolution.

D. Testing Results

The CNN discovered that features like a grin, teeth, eyebrows, and eye widening trigger it, and that each feature remains constant within the same class. These findings show that the CNN has learned to view human-like characteristics that have generalized elements. These interpret able results point to common erroneous classifications, such as people wearing glasses being labeled as "angry." This occurs because when it thinks a person is frowning, the label "angry" is highly triggered, and frowning features are confused with darker glass frames. Furthermore, the features learned in the mini-Xception model are more interpret able than those learned in the sequential fully-CNN model. As a consequence, using more parameters in implementations results in less stable functionality. The architectures were designed in a systematic manner to reduce the number of parameters. It started by removing all completely connected layers and using depth-wise separable convolutions to minimize the number of parameters in the remaining convolutional layers. For multi-class classifications, the templates can be stacked while preserving real-time inferences. It has an system that combines face recognition, gender classification, and emotion classification into a single module.

II. Model 2

On open source facial datasets downloaded from Kaggle and IMDB, a custom VDCN model is used. To assist facial detection and video processing, OpenCV, dlib, and keras were used. For large-scale image recognition, very deep convolutional networks are used. To finish the processing, a VDCN

network architecture and a softmax classifier are used.The design gradually increases the network's depth by adding more convolution layers, which is possible because all layers use very small (33) convolution filters. As a result, the design generates a substantially more accurate convulsion network that performs well on ILSVRC classification and localization tasks and is also applicable to other image recognition datasets, where it can achieve excellent results even when used as part of a relatively simple pipeline (e.g. deep features classified by a linear SVM without fine-tuning).

A. VDCN, Very Deep Convoluted Network

The ConvNets are fed a fixed-size 224 224 RGB image during preparation. The only pre-processing performed is subtracting each pixel's mean RGB value from the training collection. The picture is passed through a stack of convolutional (conv.) layers, where filters with a very limited receptive field were used: 3 3 (the smallest size to catch the notions of left/right, up/down, and center) were used. 1x1 convolution filters were used in one of the setups, which can be thought of as a linear transformation of the input channels (followed by non-linearity). The convolution stride is set to 1 pixel, and the spatial padding of conv. layer input is set to 1 pixel for 3 conv. layers so that the spatial resolution is maintained after convolution. Five max-pooling layers, which obey some of the conv. layers, conduct spatial pooling (not all the conv. layers are followed by max-pooling). Max-pooling is done with stride 2 over a 2 2 pixel window.

Three Fully-Connected (FC) layers adopt a stack of convolutional layers (which have different depths in different architectures): the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus includes 1000 channels (one for each class). The soft-max layer is the final layer. In all networks, the fully connected layers are designed in the same way. The rectification (ReLU (Krizhevsky et al., 2012) non-linearity is present in all hidden layers.Local Response Normalisation (LRN), which does not boost efficiency on the ILSVRC dataset but increases memory usage and computation time, is present in none of the networks (except for one).

B. Training

The training protocol for ConvNet is based on Krizhevsky et al (2012). The training is achieved by using mini-batch gradient descent (based on back-propagation (LeCun et al., 1989) with momentum to optimize the multinomial logistic regression objective. The batch size and momentum were both set to 256 and 0.9, respectively. For the first two fully-connected layers, the training was regularized by weight decay (the L 2 penalty multiplier was set to 5.10-4) and dropout regularization (dropout ratio set to 0.5). The learning rate was originally set to 10-2, but when the validation set accuracy stopped improving, it was reduced by a factor of ten. The learning rate was reduced three times in total, and the learning was stopped after 370K iterations (74 epochs). We hypothesized that, despite having more parameters and a deeper depth than (Krizhevsky et al., 2012), our nets took less epochs to converge due to (a) implicit regularisation enforced by greater depth and smaller conv.filter sizes; and (b) pre-initialisation of some layers. The initialization of the network weights is critical, as poor initialization can cause learning to stall due to the gradient instability in deep nets. The first four convolution layers were initialized with the layers of net A while training deeper architectures, and the last three completely connected layers with the layers of net B. (the intermediate layers were initialised randomly). The learning rate for the pre-initialised layers does not decrease, allowing them to adjust during learning. The sampled weights from a normal distribution with a zero mean and 10- 2 variance were used for random initialisation (where applicable).The biases were set to zero at the start. The 224x224 ConvNet input images were randomly cropped from rescaled training images to obtain the fixed-size 224224 ConvNet input images (one crop per image per SGD iteration). The crops were randomly horizontally flipped and randomly RGB color shifted to add to the training set (Krizhevsky et al., 2012).

C. Testing and Results

Given a trained ConvNet and an input image, it is classified as follows at test time. The image is first isotropically rescaled to the smallest image side. In a similar way to how the network is applied densely over the rescaled test picture (Sermanet et al., 2014). Convolutional layers are created by first converting fully-connected layers to convolutional layers. The fully-convolutional net that results is then applied to the entire picture. The result is a class score map with a variable spatial resolution based on the input image size and a number of channels equal to the number of classes. Finally, the class score map is spatially averaged to obtain a fixed-size vector of class scores for the image (sum-pooled). The test was also enhanced by horizontal image flipping; the soft-max class posteriors of the initial and flipped images were combined to obtain the image's final scores. The tested very deep convolutional networks (up to 19 weight layers) are used for large scale image classification in this study. It was proved that the representation depth is beneficial for the classification accuracy, and that state-of-the-art performance on the ImageNet challenge dataset can be achieved using a conventional ConvNet architecture. The results yet again confirm the importance of depth in visual representations.

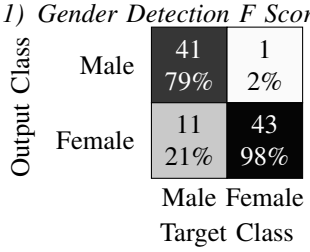
III. Evaluation Results between Model 1 and Model 2

Both the models are built and trained on very robust technologies available as open source. In general, Model 1 seems to be more pragmatic compared to Model 2. However to quantify and select a better suited model; both models were tested against random 100 images from the internet. The accuracy of the modles were judged based on the F score calculation. In statistical analysis of binary classification, the F-score or F-measure is a measure of a test's accuracy. It is calculated from the precision and recall of the test, where

the precision is the number of correctly identified positive results divided by the number of all positive results, including those not identified correctly, and the recall is the number of correctly identified positive results divided by the number of all samples that should have been identified as positive. Precision is also known as positive predictive value, and recall is also known as sensitivity in diagnostic binary classification. Below is the result of F score calculation of both the models in terms of gender and emotion classification :

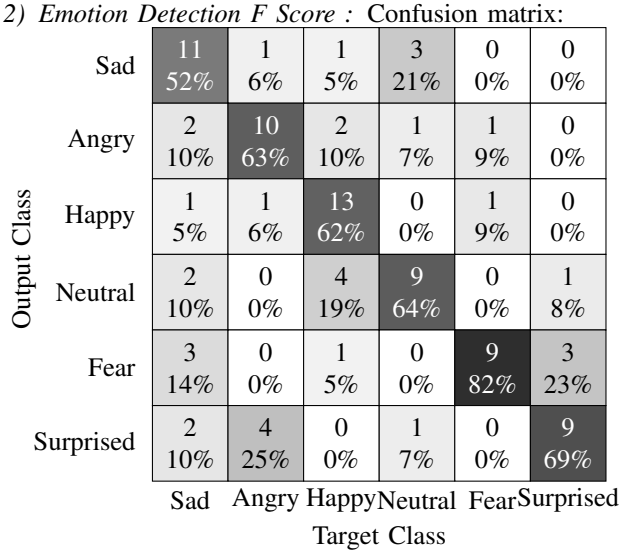
A. Model 1 Results

1) Gender Detection F Score : Confusion matrix:



Average Precision = 0.89
Average Recall = 0.89
F Score = 0.89

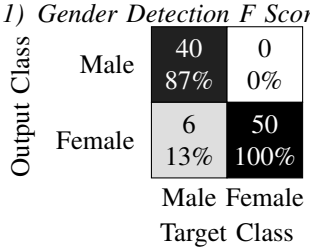
2) Emotion Detection F Score : Confusion matrix:



Average Precision = 0.65
Average Recall = 0.64
F Score = 0.64

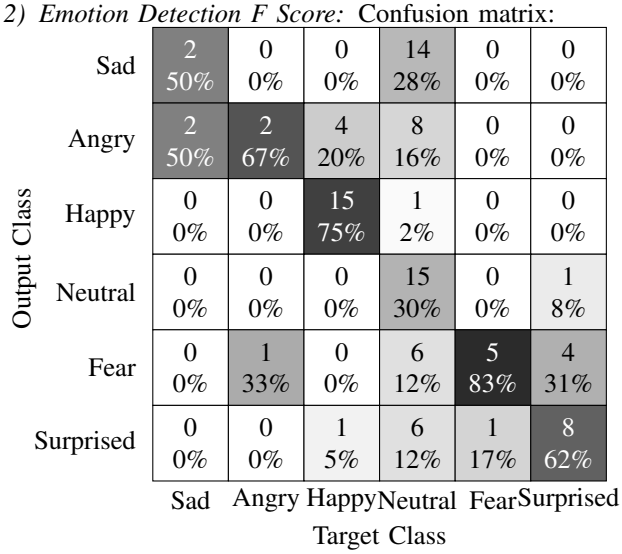
B. Model 2 Results

1) Gender Detection F Score : Confusion matrix:



Average Precision = 0.94
Average Recall = 0.95
F Score = 0.94

2) Emotion Detection F Score: Confusion matrix:



Average Precision = 0.61
Average Recall = 0.49
F Score = 0.54

IV. Conclusion

As witnessed by the f scores of each of the model's evaluation, both the models seems to be doing relatively very well with Gender detection(Model 1 = 0.89 , Model 2 = 0.94). The slightly lower score of Model 1 can be credited to the testing images with children between the age of 2-3 years, which with all honesty is capable of confusing humans too.

When it comes to emotion detection, Model 1 clearly has a better f score and this I believe is due to the more efficient approach. As it can be observed in the confusion matrix, Model 2 always classifies the emotion as 'neutral' when the confidence score isn't high enough. However, in Model 1 it follows an order of precedence and chooses the emotion with the best confidence score. One would argue over the risk of classifying a 'sad' face as 'happy' is one to avoid but in terms of the position I would like to implement this in, a more pragmatic approach suits better.

V. Future work

I plan to integrating ROS services along the Model 1 and create a ready to run docker image in the college netwrok such that emotion and genders of human can be detected by Tiago Robot.

References

[1] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In NIPS. MIT Press, 2006. 2, 3

[2] D. R. Wilson and T. R. Martinez. The general inefficiency of batch training for gradient descent learning. Neural Networks, 16(10):1429–1451, 2003. 4

[3] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination and expression (PIE) database. In In Proc. FG, 2002. 2

[4] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. CoRR, abs/1412.1265, 2014. 1, 2, 5, 8

[5] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In Proc. ECCV, 2014. 7