Enhancing Human Face Recognition with an Interpretable Neural Network

Timothy Zee, Geeta Madhav Gali, and Ifeoma Nwogu
Rochester Institute of Technology
1 Lomb Memorial Drive, Rochester NY
{tsz2759; gg6549; ionvcs}@rit.edu

Abstract

The purpose of this work is to determine if the ability to interpret a convolutional neural network (CNN) architecture can enhance human performance, pertaining to face recognition. We are interested in distinguishing between the faces of two similar-looking actresses of Indian origin, who have only a few discriminating features. This recognition task proved challenging for humans who were not previously familiar with the actresses (novices) as they performed only just better than random. When asked to perform the same task, humans who were more familiar with the actresses (experts) performed significantly better. We attempted the same task with a Siamese CNN which performed as well as the experts. We therefore became interested in applying any new knowledge obtained from the CNN to aid in improving the distinguishing abilities of other novices. This was accomplished by generating activation maps from the CNN. The maps showed what parts of the input face images created the highest activations in the last convolutional layer of the network. Using "fooling" techniques, we also investigated what spatial locations on the face were most responsible for confusing one actress for the other. Empirically, the cheekbones and foreheads were determined to be the strongest differentiating features between the actresses. By providing this information verbally to a new set of novices, we successfully raised the human recognition rates by 11%. For this work, we therefore successfully increased human understanding pertaining to facial recognition via post-hoc interpretability of a CNN.

1. Introduction

In this work, we are interested in understanding how a convolutional neural network (CNN) would work to differentiate between two similar subjects, having only a few discriminating features. The subjects we employ are the faces of two famous Bollywood actresses who have acted in a large number of films and TV series, thus providing a rich source of data for our evaluation. They are Aishwarya Rai and Priyanka Chopra (referred to for the rest of the document as AR and PC, respectively). Their faces are similar enough that humans unfamiliar with them (novices) only do slightly better than chance in differentiating them.

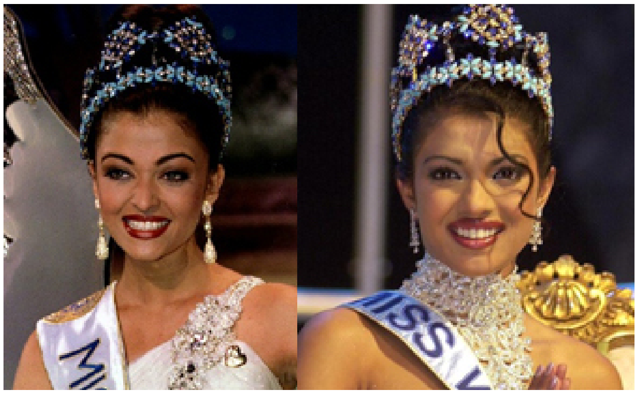


Figure 1. Are these 2 images of the same person or of different people? If different, what distinguishing features can you observe? If same, what features are most similar? This recognition problem is the basis of this paper. (Image best viewed in color or online).

We are interested in measuring how well a CNN can tell the difference between these two actresses and then compare with how well novices perform on the same task before and after they are influenced by the CNN on where to look. This provides empirical evidence that CNNs can potentially assist human learning for facial recognition tasks. We should note however that for this work, we focused on only two actresses since we are more interested in transferring knowledge from the CNN to humans, rather than in general facial recognition.

The purpose of this work is to move us closer to understanding the potential uses of neural network interpretability in aiding humans with hard prediction tasks. For this rea-

2473-9944/19/$31.00 ©2019 IEEE
DOI 10.1109/ICCVW.2019.00064

son, we selected a Siamese CNN, which allows us to effectively train a model useful for learning their differentiating features, when presented with pairs of input images.

The resulting Siamese CNN heatmaps show the spatial locations in the input images where the CNN has the highest positive activations in the last convolutional layer. This allows us to determine the regions of the input images where the network focuses to create its decision boundaries between the two actresses. This information is then presented to novices in order to increase their differentiating abilities.

We then use a fooling network which makes slight adjustments to images to fool the CNN into thinking an image of one of the actresses is actually the other, and predicts this with high confidence. The modifications to the images are restricted to be small enough that humans cannot distinguish between the pre-fooled and the post-fooled images. This allows us to examine the noise that is created when taking the post-fooled image and subtract it from the pre-fooled image. The difference in these images is how the fooling network manipulated the image to trick the CNN into misclassifying that image. This visualization allows us to see which regions in the image the fooling network is using to trick the network.

The main contribution of this paper is providing empirical evidence towards the utility of post-hoc deep neural network interpretability methodologies for enhancing human understanding when dealing with difficult classification problems.

2. RELATED WORK

CNNs have demonstrated excellent performance at image recognition (and other related computer vision tasks such as scene classification [17], image segmentation [6], etc.), starting from their earlier versions by LeCunn *et al.* [4]. Li *et al.* demonstrated the efficacy of CNNs for face detection at multiple poses, on a large real-world dataset consisting of 5,171 annotated face images. In the same year, Parkhi [7] *et al.* successfully presented the use of the VGG16 and 19 networks for use in face recognition from a single image or tracked from videos, given a very large training dataset. Data availability for training face recognition networks is still challenging in general, since many very large-scale datasets are proprietary and owned by companies like Facebook [12, 13] and Google [9].

In recent years, many new complex and significantly deeper convolutional networks have emerged. These include the GoogleNet Inception network [11]. GoogleNet included an inception module which was significantly different from all the previous sequential architectures by having several components of the network happen in parallel. The Microsoft ResNet architecture is an extremely deep CNN structure with 152 and growing number of layers, consisting of residual blocks. Although these more advanced networks have been shown to perform better than traditional architectures, and we would probably have obtained significantly better recognition results with them, for interpretability purposes, we stick to a traditional convolutional architecture for our Siamese model, described in more detail in Section 4.

For gaining deeper insight into the internals of the network, different visualization methods have been proposed such as by Zeiler and Fergus [15], where a convolution-deconvolution network is used to map activations at different layers in the network back to the image pixel to determine what pattern in the input image caused a given activation in the feature maps. The behaviors of the network at the higher activation layers provide insight into the features that the topmost layers extract before passing the representations to the fully connected dense layer(s).

For the rest of the paper, in Section 3 we briefly describe the data collection and cleaning processes, we present the CNN model used for the recognition task in Section 4; in Section 6 we discuss the recognition tests performed on humans and discuss what we observe from the internals of the network in Section 5, which deals with visualizing input regions that maximize activations. Lastly, we discuss our findings and conclude in Section 7.

3. DATA COLLECTION AND PRE-PROCESSING

As mentioned earlier, we selected two Indian actresses AR and PC who are well known for their acting accomplishments. AR won the Miss World contest in 1994 and PC won it in 2000 and between them have starred in more than 300 full feature films, thus providing us with a rich source for collecting their face data online. Another major reason for choosing them for this study is that, though unrelated, they look quite similar making it quite difficult for a person (especially one not familiar with Bollywood movies[1], whom we will refer to as novices) to consistently distinguish between AR and PC when seeing their pictures for the first time.

We collected the data for this work by searching for a specific movie where one of the actresses was a lead between 2008 and 2016, and downloading all the images associated with the movie. This process was repeated for about 30 - 40 films for each actress. The actresses rarely act together and we did not encounter situations where they co-appeared in the same movies or images. Many images extracted from these movies had faces of other actors appearing alongside AR or PC. The data cleaning process involved:

1. Running a state-of-the-art face detector to extract all

[1] In recent times though, the actress PC has expanded her career to the US, making her more internationally known.

faces in all frames/images obtained from online

2. Cropping and aligning all the faces from the images
3. Resizing the images to 128×128 as most cropped images obtained were this size or larger.
4. Removing the images that are not either AR or PC involved a 2-step process:
   (a) Step 1: We built a 3-layer neural network trained with the correctly annotated images of AR and PC and the output of the network was a Softmax classifier which the model categorizes all the images in three categories - AR, PC and Other. This process was not very effective in removing noisy images because it was heavily skewed towards the positive classes. The negative class data used in training did not fully represent the whole set of faces of the other actors who acted with AR and PC. When the class score provided by the Softmax classifier exceeded 90%, we safely accepted this result as the correct identity of the actress.
   (b) Step 2: We then manually went through the remaining images and deleted as many noisy faces as possible resulting in a total of 2,275 images of AR and 2,295 images of PC.

The total data size is 4,570. The data was then divided into 3 categories - training, validation and testing. A total of 1000 were removed for validation and the remaining data was split in the ratio 2.5:1 training to testing.

4. THE RECOGNITION MODELS

We use two similar techniques for learning the identities of the two actresses-under-investigation. The first network is purely for recognition so that given any input image in testing, it returns a value indicating whether the image is of AR or PC. The second architecture attempts to directly mimic the human tests which is presented in more detail in Section 6. In this network, two images are simultaneously presented and the network attempts to identify whether it is the same individual in the images or different individuals. The architectures and training schemes are described below.

4.1. The Basic CNN

The CNN used for this work is a slight variant of the well-known VGG16 model showed in Figure 2, as the VGG family of models has empirically shown to learn well when dealing with face image data. The network consisted of 16 weight layers, using very small (3×3) convolution filters on the the layers, and (2×2) pooling intermittently as shown in Figure 2a. After each convolution layer, rectified linear unit (ReLU) activation is performed, although that is not shown in the diagrams for clarity purposes. The last convolution layers is flattened and fed into a fully connected network whose layers are also referred to as the dense layers. The last fully-connected layer is fed into a softmax classifier.

The VGG19 network is very similar to VGG16 but has additional convolution layers. We tested various architectures on our recognition task and found a structure with 21 layers (instead of the more popular ones with 16 or 19 layers) to perform the best on the task at hand. While the input image size to VGG16 is 226×226, we use 128×128 due to the nature of the images we collected. Our modified variant of the network is shown in Figure 2b.

4.1.1 Training the Network

We present the set of hyper-parameters the yielded the best accuracy in training the network on the actresses' data:
Dropout: We trained the network using dropout values at 0.2, 0.25, 0.3, 0.35, 0.4; **0.3** gave the best result in comparison to all other drop out values.
Batch size: We trained the network in batches with a size of 30 as it maximized the GPU resources available to us.
Number of Epochs: We trained the network using 125, 150, 200 epochs and settled for 200 as the most efficient given our resource limits.
Learning rate: We trained the network with learning rates of e-5, e-6, e-7 and found e-7 giving the best results without taking excessively longer time to train.
Batch normalization: Using batch normalization thrice in the course of the training gave us better results. Batch normalization was applied on the output of each of the fully connected dense layers of the neural network.

The best and final accuracy given by the network for differentiating between the two actresses was **87.3%** on our test set of 1000 images. The average accuracy and average loss for the training and the validation data can be seen in Figures 3 and 4 respectively.

4.2. Fooling the Network

With a technique similar to that proposed by Karpathy on his blog [2], we implemented a routine which adjusted the gradients of the network in relation to the image pixels, to "fool" the network into wrongly classifying an object with high confidence. The original image was altered by iteratively adding some noise to its pixels applying the constraint that kept a 15% limit on how much each pixel could change, and then we attempted to maximize the error of the model. The resulting pictures looked the same to humans but the network now assigned them with $\geq 90\%$ probability to the wrong class. Some results from fooling the network are shown in Figure 5.

To visualize what the difference is between the pre-fooling and post-fooling images, we created a visualization method

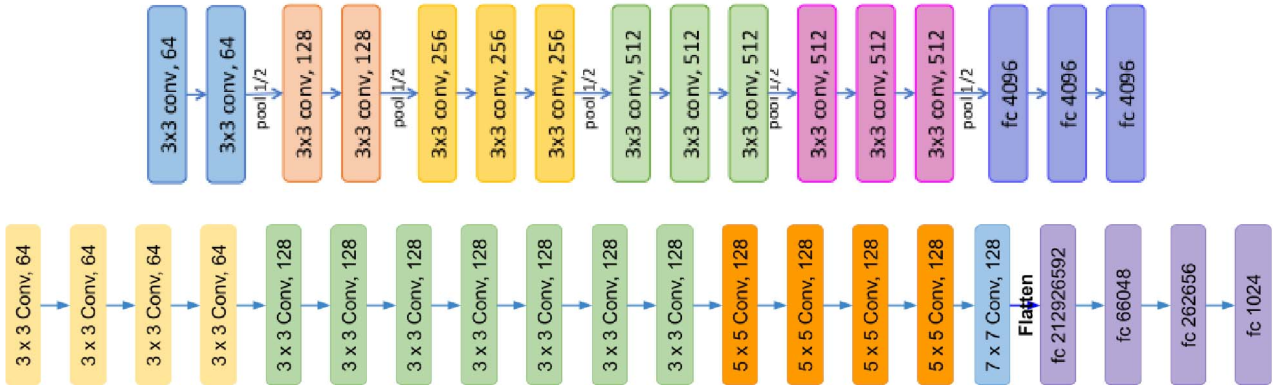


Figure 2. The VGG16 network (top) and our modified variant (bottom) used for the recognition task.

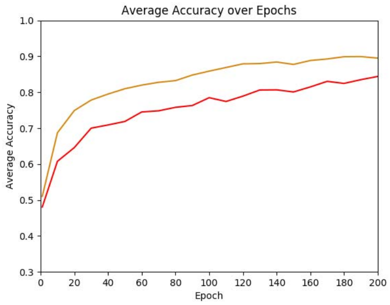


Figure 3. Training and validation accuracy of the model over 200 epochs.

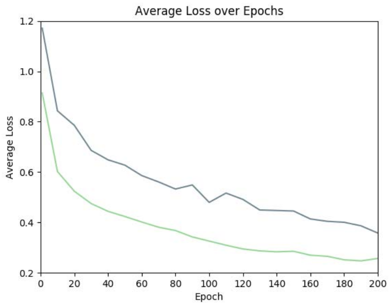


Figure 4. Training and validation loss of the model over 200 epochs.

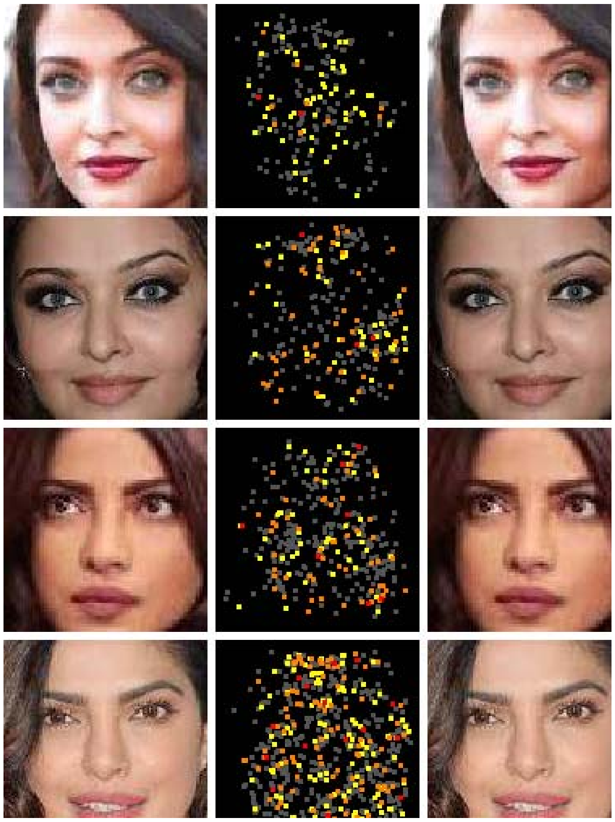


Figure 5. The top two images show results of fooling the network into thinking that images of AR are PC and the bottom two are the reverse, PC images fooled to being classified as AR. The leftmost image is the original, the rightmost image is the altered version and the middle is the enhanced visualization of the difference between them, resulting from the adjusted pixels.

that focused on finding the largest changes to pixel intensity when the post-fooled image is subtracted from the pre-fooled image. To do this we first convert the noise image to grayscale and then create a basic pixel-map image on a pixel-by-pixel basis to visualize the biggest changes from a pre-fooled image to a post-fooled image. To remove noise in the image, if any pixel had a grayscale value less than 150, we visualized it as black. We then show more significant background changes in gray. We found that pixel values between 150 and 180 best represented general changes to the face.

Beyond that we used red, orange, and yellow pixels to show the varying importance of pixels, with red being the most significant. Figure 5 shows the result of the adjusted changes that resulted in misclassification of the original images.

Table 1 shows the pixel-map color scheme we used. This allowed us to easily see the most important changes the fooling network made to the images. Closest neighboring pixels of pixels with intensities above 150 are shown with the same color to enlarge these changes between the images.

Intensity Value of a Pixel(I)	**Color**
$I \leq 150$	Black
$151 < I \leq 180$	Gray
$181 < I \leq 200$	Yellow
$201 < I \leq 225$	Orange
$226 < I \leq 255$	Red

Table 1. The colors chosen to represent the ranges of intensity for the difference between the post-fooling image pixels and pre-fooling image pixels.

We find with the visualization of the differences between the post-fooling and pre-fooling images, that several parts of the face seem to be consistently highlighted including the forehead, the cheeks, and lower portions of the face excluding the mouth. We note the eyes seem to have no real changes from the pre-fooled images to the post-fooled images, indicating the eyes are not a good feature to trick the network from misclassifying one actress as the other. These highlighted areas provide an insight on how the network changes its classification decisions based on certain regions of the images. This allows us to find the most influential features the network looks for when differentiating between the two actresses.

4.3. The Siamese Network

The next type of CNN used in this work is a 10 layer Siamese model that takes a pair of images as input. Due to the nature of the data collected, the images are resized to 128×128. Each image in the input pair propagates through 10 weight layers before being flattened and sent through an average pooling layer which is then joined in the first of two fully connected layers. A softmax classifier is applied after the last fully connected layer. The model has very small filters, starting with 2 layers of 64 (3×3) filters followed by 8 layers of 128 (3×3) filters. Pooling is applied intermittently as shown in Figure 6. After each convolution layer, rectified linear unit (ReLU) activation is preformed, although not shown for clarity purposes. To perform weight updates, a modified version of binary cross-entropy is used. Instead of traditional binary cross-entropy where the loss $L = -y\log(p) + (1-y)\log(1-p)$; here, the loss is computed by first computing a batch of positive samples, and then negative samples individually, notated as L_+ and L_-, respectively. These values are summed to result in a complete loss function of $L = L_+ + L_-$. Our complete model is shown in Figure 6.

For both training and testing of the Siamese model, we take our original training data and create equal pairs of positive samples (a pair of the same actresses) and negative samples (pairs of both actresses) and create a balanced dataset of positive and negative pairs of examples for training and testing. The best and final accuracy given by the network for differentiating between the two actresses was **85.04%** on our test set of 2000 images.

5. CNN CLASS ACTIVATION MAP

Class activation maps (CAMs) [16, 3] are attention maps that indicate the most discriminative regions used by a CNN to identify a specific category/class. The CAM for a class can be viewed as the weighted sum over the feature maps of the last convolutional layer. A global average pooling layer is used to convert the feature map into a single value, and then used for calculating the associated weights. CAMs naturally allow for re-using classifiers for localization, even when training without any bounding box coordinates data. This suggests that CNNs have some kind of built-in attention mechanism.

Guided backpropagation [10] propagates only positive gradients as it sets any negative gradients that are in the current layer and the previous layer to zeros. As gradients are computed through the layers of the network, this technique only highlights nodes that that have entirely positive gradients through the network, thus making it more accurate than the basic backpropagation algorithm, which only considers one layer at a time. We use guided backpropagation to generate the CAMs for our recognition network.

Figure 7 shows a set of heatmaps corresponding to the activation maps generated when recognizing that actresses are the same person (shown in columns 1 and 2) and when the input images are of the different actresses (shown in column 3). The heatmaps in Figure 7 show which areas of the input images contribute towards maximizing the output filters of the last convolution layer for each class. When examining two images of the same actress (as in columns 1 and 2) we apply the "same" filter for the heatmap which shows us where the two images activated the most for the "same" class. Similarly, for the "different" class where both actresses are present in the input (as in column 3), we apply the "different" filter.

This allows us to visualize the network's attentiveness to different areas of the face. By knowing where the network is attending to make its decisions from the input images, we can inform users, thus aiding in their abilities to distinguish between the actresses.

With this technique, we consistently find that certain areas of the input images are important for maximizing the filter activations for both cases where the input image pairs are of the same actress, and when they are of different actresses. We note that the forehead, cheeks and nose are

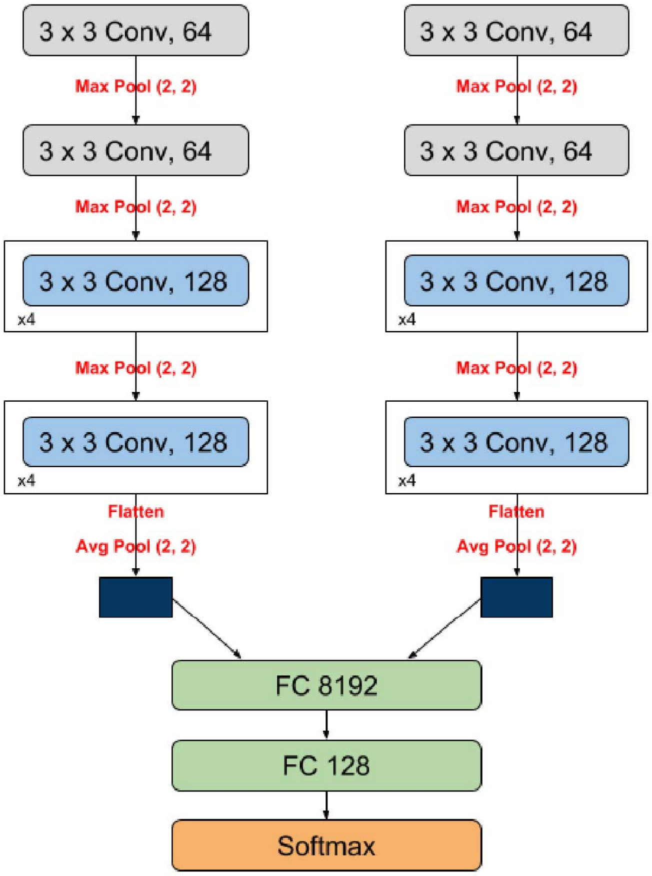


Figure 6. Siamese network model propagates each input image through a 10 layer convolutional network segment that joins together in the fully connected layers where softmax activation is applied. Note that some layers repeat $4\times$.

the most significant areas when considering heatmaps of the same actress. From the fooling noise visualizations of the standard CNN, along with the Siamese CNN class activation maps, we conclude the cheek bones and forehead may be the best distinguishing factors of the two actresses for the network.

6. HUMAN TESTS

In 2011, Carbon [1] showed that processing of faces (matching of sequentially presented faces), was possible when faces were presented for only 100 milliseconds or less. To overcome any biases in recognition, in our work, we increased the processing speed to 400-1200 ms.

To compare the Siamese CNN results with human performance, we conducted a survey where we first presented novice participants with twenty (20) different pictures of one of the actresses, followed by another 20 pictures of the other actress at intervals of 400 milliseconds (ms). This is done to "train" the participant on what each actress looks like at the 128×128 resolution. Figure 8a is the screen first presented the participants to "train" them on what each actress looks like at the 128×128 resolution.

Once the training session is completed, a different screen shown in Figure 8b is presented to the participant: a pair of images is flashed to the user for a 1200 millisecond duration. The user is then expected to select a button indicating whether the images displayed were of the same person or not. Twenty different pairs of images are shown to the participant. This is a slightly easier task than asking the participants to select the identities of the actresses since they might not necessarily have memorized the labels assigned to each actress, even if the participant can distinguish between them. Asking the participants to determine if the images are of the same actress or not, still requires recognition, without the need for label assignment. This is similar to the task of the Siamese network. The user scores a point if he/she correctly determines if the pair of images presented belong to the same actress (`Yes- Same`) or not (`No - Diff`), and scores a zero if he/she is wrong. The accuracy for M participants is computed as:

$$\text{accuracy} = \frac{\sum_i^M \text{Correctly classified by participant } i}{M \times \text{Total pairs shown (N=20)}} \quad (1)$$

The survey was conducted on three sets of participants. In the first survey, we randomly selected 200 images of each actress and used them to run the test described above. The

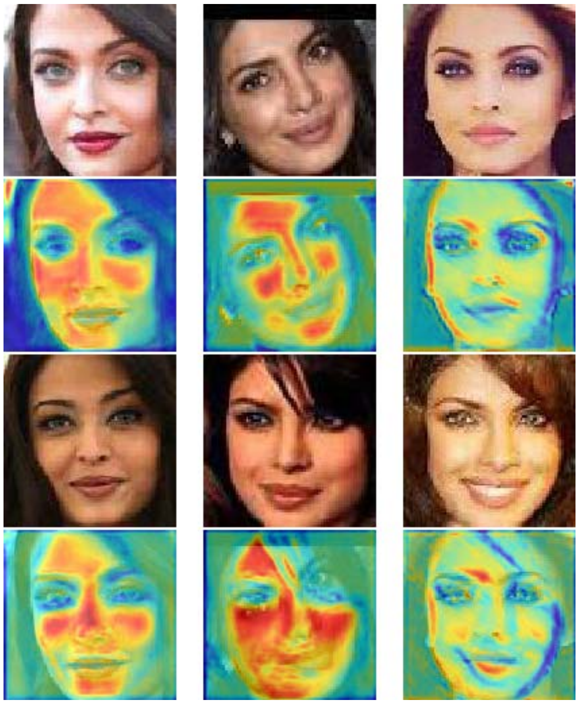
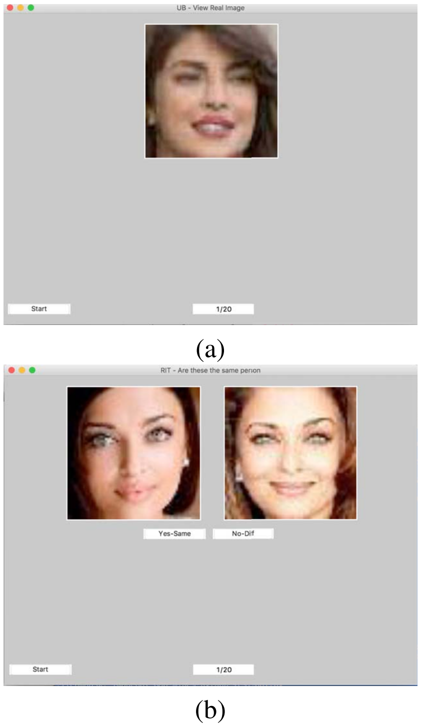


Figure 7. Heatmaps for the “same” class (first two columns) and “different” class (last column) where the first (top) row of images and third row of images correspond to the pair of images passed into the network, and the second row and forth (bottom) rows of images are the heatmaps corresponding to that pair of images.

images that each participant was surveyed on were random out of the selected 200 images, and there was an equilikely chance of getting a ground truth value of the same actress or different actresses. There were 15 participants for this test, none of which claimed were familiar with the actresses (novices). We call this the *pre-test*. The accuracy of this group was just about random at **52.67%**

For the second survey, which we call the *post-test*, we used the same 200 images used in the pre-test to generate random pairs and applied the same process again, but this time instructing the participants to focus on the *forehead and cheekbones*, to distinguish between the actresses. There were 15 different novice participants for the second test. No subjects of Indian origin participated in the pre- and post-tests as they would have been more likely to be familiar with the actresses or show cultural biases. The accuracy of this group was **63.67%**.

The third group consisted of 5 participants who were of Indian origin, and familiar with both AR and PC (experts). They were not given any focusing instructions, much like the pre-test. The accuracy in differentiating between the two actresses in this group was **85.0%**. This allowed us to examine the experts' performances, even if they could not explain precisely how they successfully distinguished the two actresses.



(a)

(b)

Figure 8. The top image shows the screen used to train the participants on recognizing each actress individually; the bottom image shows the screen used to test the participants on their recognition capabilities.

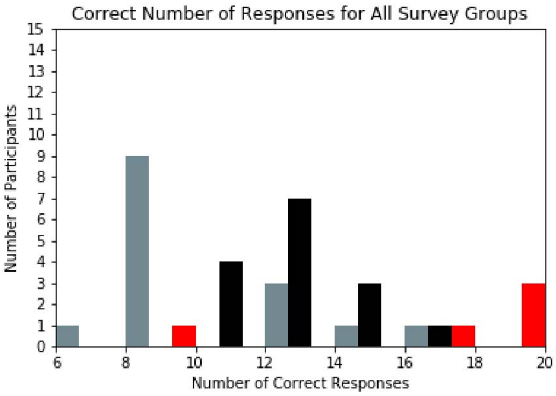


Figure 9. Histogram results for our three survey groups. The number of correct responses is shown left for the pre-test (gray), middle for the post-test (black), and right for the experts (red). Some pre-test novices scored as low as $\frac{6}{20}$ and some experts as high as $\frac{20}{20}$. No post-test novice scored lower than $\frac{11}{20}$.

Figure 9 shows the distribution of the results from the 3 surveys. Note that the last survey only involved 5 expert participants unlike the others which had 15 participants each.

7. DISCUSSION AND CONCLUSION

We designed a couple of CNN architectures which significantly outperformed *novice* humans in recognizing two famous and similar looking Bollywood actresses. While humans taking the pre-survey performed only slightly better than random, the network yielded an accuracy value $> 85\%$, equivalent to the *expert* humans.

We therefore used class activation maps to better understand what the CNN paid attention to in the inputs, to yield its decisions. We then visualized the regions of the inputs which maximized the activations of the network's last convolutional layer. This allowed us to interpret where in the input the best distinction locations between the two actresses might be.

We further examined the utility of a fooling network to visualize the modifications it makes to images to cause the network to misclassify, thus providing us with the differences the network is looking for when distinguishing between AR and PC. The pixels-maps shown in Figure 5 are consistently higher in similar areas as discovered by the activation maps, thus empirically demonstrating that the network pays particular attention to specific areas of the face during the recognition task.

We found that for our dataset, the cheekbones and foreheads appear to be the main sources of differences when trying to distinguish between AR and PC. Interestingly, we also found that the Siamese CNN appeared to avoid the eyes and mouth. This finding was rather unintuitive as we initially expected the areas of strong texture such as the eye or mouth regions, to be the most distinguishing.

We used the results from this to survey people, to determine if providing them with information learned from interpreting the CNN would guide novice participants to better distinguish between the two similar actresses and found an improvement of about 11 percentage points.

In conclusion, we have provided empirical evidence demonstrating that for a basic (but nontrivial) binary classification task that both a traditional and a Siamese CNN can find features that humans are able to use to assist in distinguishing between two similar people. The CNN performed right on-par with expert humans and elevated the accuracy rate for novices.

References

[1] C.-C. Carbon. The First 100 milliseconds of a Face: On the Microgenesis of Early Face Processing. *Perceptual and Motor Skills*, 113:859–74.

[2] A. Karpathy. Breaking Linear Classifiers on ImageNet, 2015.

[3] R. Kotikalapudi and contributors. keras-vis, 2017.

[4] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, and L. Hubbard, W. J. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[5] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5325–5334, June 2015.

[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.

[7] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference - BMVC*, 2015.

[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[9] F. Schroff, D. Kalenichenko, and J. Philbin. acenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[10] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity : The all convolutional net. pages 1–14, 2015.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[12] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2746–2754, 2014.

[13] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Webscale training for face identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2746–2754, 2015.

[14] L. van der Maaten and G. Hinton. Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research (JMLR)*, (9):2579–2605, 2008.

[15] M. D. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. In *European Confrence on Computer Vision (ECCV)*, volume 8689, pages 818–833. Springer International Publishing, 2014.

[16] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. *CVPR*, 2016.

[17] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 487–495. Curran Associates, Inc., 2014.