Facial Attributes Detection Using CNN Deep Learning Algorithm

Anirudhan Anbuchezhian
School of Computing and Mathematical Sciences
(of Affiliation)
University of Greenwich
London, United Kingdom
aa0854y@greenwich.ac.uk

***Abstract*— Facial recognition has improved drastically due to technology improvement in the recent years. This has been possible with the power of deep learning. Recent studies have shown that Convolutional Neural Network, a class of deep learning, has shown a high accuracy in predicting face detection. In this paper, Convolutional Neural Network (CNN) will be used to detect facial attributes of different images. VGG16, a CNN architecture, is going to be used as a branch architecture to build a network for each attribute. The performance of the models is evaluated using Accuracy, Precision, Sensitivity and F1 Score.**

Keywords—CNN, VGG16, deep learning, facial attributes.

I. INTRODUCTION

When it comes to human faces, Facial Recognition has always been one of the most interesting and intriguing developments. With recent developments in image processing recognition technology, various face attributes, such as wrinkles, freckles, hair colors, hair thickness and glasses can be detected by image recognition technologies. Convolutional Neural Network (CNN) is a powerful deep learning network for image recognition, that gained its popularity in recent years. CNN, which is a class of deep learning, is used for creating models that can detect various facial attributes for facial recognition.

In this paper, CNN, VGG16 architecture, is going to be used to create different models for detecting each attribute. The performance of the models is going to be evaluated by using confusion matrix. The dataset consists of images of different faces, with various facial attributes.

The rest of the paper is organized as follows. Chapter II provides a summary of related articles towards the prediction of facial recognition using CNN . Chapter III shows the data collection, preparation and cleaning process. Chapter IV gives a description of the model that is created and how the performance of the model is measured. Chapter V shows the results produced by each model with their respective performance. Chapter VI and VII provides a conclusion for the finished experiments and conveys how it could be improved more in the future.

II. RELATED WORK

Chin et al (Chin et al, 2018) developed a system to detect facial pores in various shapes using CNN (Convolutional neural network). The reason for them to develop a system using CNN is because traditional image processing makes it more difficult to recognize facial pores, due to the small size of pores. They created three different models where LeNet-5 was their benchmark architecture.

Jiang and Learned-Miller (Jiang and Learned-Miller, 2017) created models using Faster-CNN to demonstrate face detection performance on three benchmark datasets. Their experiment showed impressive results using Faster R-CNN for face detection, even though it was designed for generic object detection. Their experiment also showed that multiple convolutional layers within RPN (Region Proposal Network) can be used without any computational burden, due the sharing of convolutional layers between the RPN and Fast R-CNN.

Basbrain et al (Basbrain et al, 2017) developed a model using shallow CNN architecture called shallow-GlassesNet to detect eyeglasses. Shallow CNN consists of just six layers, three of which are convolution layers, two of these are max-pooling layers and one is a fully connected layer. The dataset image that they used was a pre-trained GoogleNet which was adjusted with images that had both glasses and non-eyeglasses. The trained weights from GoogleNet were used in the shallow-GlassesNet layers. Their model showed high precision and high speed for detecting eyeglasses detection which made it ideal for use in real-time applications.

Zhong et al created (Zhong et al, 2016) models using CNN Google's Facenet and VGG architecture. They tried to predict face attributes using a pre-trained CNN for face recognition. Their experiments ran on two large datasets which were LFWA and CelebA. Their model predicted facial attributes by properly leveraging CNN.

III. DATA PREPARATION

The dataset required to build a face attribute detection system should consist of facial images of different varieties such as age, gender, ethnicity, emotions, place etc. By using different facial images, face attributes detection system can enhance its recognition rate. This will enable the system to detect accurate attributes.

A. Data Collection and Preparation

For the detection system, 1999 facial images are gathered to use as training and test datasets. These images are

merged into a single video, which will be fed into an attribute annotation tool provided by the university, to detect the attributes for each image manually as a data label, that are binary values. The binary values for each attribute are:

- wrinkles (binary: has/does_not_have), class 0: does_not_have, class 1: has
- freckles (binary: has/does_not_have), class 0: does_not_have, class 1: has
- glasses (3 values: do_not_wear/wear_normal/wear_sunglasses), class 0: does_not_wear, class 1: wear_ normal, class 2: wear_sunglasses
- hair_color (9 values: brown/black/gray/blond/red/white/mixed/other), class 0: brown, class 1: black, class 2: gray, class 3: blond, class 4: red, class 5: white, class 6: mixed, class 7: other, class 8: not_visible
- hair_top (4 values: bald or shaved, has_few_hair, has_thick_hair), class 0: bald or shaved, class 1: has_few_hair, class 2: has_thick_hair, class 3: not_visible

These images and data labels are later imported into the system.

B. Data Cleaning

Once the dataset that consist of images and data labels is imported into the system, the data labels are cleaned, to clear all the irreverent texts in the data labels such as 'N/A', 'false' and etc. Once the irreverent words are cleared, only the relevant values are extracted, which are wrinkles, freckles, glasses, hair color and hair top.

IV. METHODOLOGY

A. Pre-Preocessing and Data Parttition

The dataset images are first resized into 96x96 input shape. This will increase the training and validation accuracy when the data is resized into higher resolutions. Two different 3-dimentional matrixes are then created. First matrix will consist of normal images without any cropping. The second matrix will consist of images with detected face from the dataset images, using cascade classifier that is available from the imported library cv2. This detected face is cropped and added to the matrix. An image from each matrix is shown in the figure below.

Figure 1: the left figure shows the first matrix with no cropping of the images and the right image show the second matrix with image cropped with detected face

Once the dataset images are pre-processed, they will be split into a training and test dataset, before creating the model. This splitting is essential to evaluate the performance of the model. 80% of the dataset images and data labels will be stored as a training set and 20% of the dataset images and data labels will be stored as a test set. This splitting is done by using the 'train_test_split' function import from the library 'sklearn.model_selection'. When splitting, shuffling is set to false to stop the randomizing when allocating datasets to training and testing. These dataset images are then normalized between 0-1. This normalizing step is important so that each input parameter has a similar data distribution and it also makes the convergence faster when training the network.

B. Deep Convolutional Neural Networks Models

CNN (Convolutional Neural Network) is a neural network with one or more convolutional layers that are primarily used for image recognition, detection and segmentation, as well as other autocorrelated data (Thomas, 2019). One of the key reasons that the world has woken up to the effectiveness of deep learning is the efficiency of convolutional nets in image recognition (Chris Nicholson, 2021). There are many researches done in detecting facial expression and image classification, where CNN has demonstrated impressive results. There are now many image recognition systems based on the CNN.

In this paper, CNN is used to create models that detect various facial attributes in an image, such as wrinkles, freckles, glasses, hair color and hair top. VGG16, a CNN architecture, is going to be used to create models for each attribute. VGG16 architecture has 16 layers, where 13 of these are convolutional layers and the other 3 are fully connected. Data augmentation is going to be applied, when creating the model to avoid overfitting. There will be 5 different CNN networks, which are created by using the VGG16 architecture for predicting each facial attribute. Predefined weight called 'Imagenet', which consists of hand annotated images, are going to be used for creating a model. In the training phase, the batch size is set to 32 and the learning rate is 0.0001. 'SparseCategoricalCrossentropy' is used as the loss function in the model to update the weights. Adam optimizer, which is an algorithm for stochastic gradient decent for training deep learning models, is going to be used to adjust the parameters during the training phase.

Adam incorporates the best features of the AdaGrad and RMSProp algorithms (Brownlee, 2021).

C. Evaluation Metrics

The evaluation of the performance achieved by the models for each attribute, can be seen by viewing accuracy, F1, precision and sensitivity scores. The performance of the accuracy, precision, sensitivity and f1 scores can be understood from a confusion matrix. A confusion matrix is a method of summarizing a classification algorithm result. The figure below shows the confusion matrix.

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

Figure 2: Confusion Matrix

From the figure above, accuracy, precision, sensitivity and f1 scores can be calculated by using the confusion matrix. The description of these metrics is shown below.

1. Accuracy

For a good classifier, Accuracy should be closer to 1. The formula for precision is shown below.

$$Accuracy = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}}$$

2. Precision

For a good classifier, precision should be closer to 1. The formula for precision is shown below.

$$Precision = \frac{\text{True Positive}}{\text{True Positive} + \text{True Negative}}$$

3. Sensitivity (aka Recall)

For a good classifier, recall should be closer to 1. The formula for recall is shown below.

$$Recall = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

4. F1 Score

For a good classifier, F1 Score should be closer to 1. The F1 Score is a greater measure than accuracy, since it is the harmonic mean of precision and recall. The formula for F1 is shown below (N B, 2021).

$$F1\ Score = 2 \times \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

V. Evaluation & Results

A. Development Environment

The experiments where training and testing of the model is done in Google Colab which takes advantage of the GPU. The machine that is used to run the experiment is a 2.6 GHz Dual-Core intel Core i5 with 8GB RAM and has Intel Iris 1536 MB Graphics.

B. Experiment Results

1. Experiments for wrinkles detection

The results shown below are for detecting wrinkles in an image. The matrix images, with the detected faces that are cropped, are going to be used for detecting wrinkles. This is for reducing the background noise, which can be found in the images.

Table 1: Results produce from training the model for wrinkles using VGG16 architecture

CNN Architecture	Accuracy	Sensitivity (aka Recall)	Precision	F1 Score
VGG16	0.7975	0.7975	0.7980	0.7976

From the table, VGG16 model performed good with testing accuracy of 79.75%.

2. Experiments for Freckles detection

The results shown below are for detecting freckles in an image. The matrix images, with the detected faces that are cropped, are going to be used for detecting freckles. The cropping is used again for reducing the background noise.

Table 2: Results produced from training the model for freckles using VGG16 architecture

CNN Architecture	Accuracy	Sensitivity (aka Recall)	Precision	F1 Score
VGG16	0.995	0.995	0.9900	0.9925

From the table above, the VGG16 model performed really high with a testing accuracy of 99.5%. Even though the percentage is high, the detection of freckles in an image is not accurate. This is due to the fact that the freckles are only seen on 18 of the images out of 1999 image dataset, which is a low dataset for freckle detection.

3. Experiments for glasses detection

The results shown below are for detecting glasses in an image. The matrix images, with the cropped faces are going to be used for detecting glasses. Again, the cropping is used for reducing background noise found in the images.

Table 3: Results produced from training the model for glasses using VGG16 architecture

CNN Architecture	Accuracy	Sensitivity (aka Recall)	Precision	F1 Score
VGG16	0.9775	0.9775	0.9670	0.9721

From the table above, the VGG16 model performed very good with a testing accuracy of 97.75%. The model was

able to detect faces with glasses or no glasses correctly, but had trouble with detecting sunglasses in certain situations.

4. Experiments for Hair colour detection

The results shown below are for detecting Hair color in an image. The matrix images with no cropping are used to detect hair color. This is due to the fact that face cropping removes some of the hair in the face detection. This could confuse the model when detecting images with no hair present.

Table 4: Results produced from training the model for hair color using VGG16 architecture

CNN Architecture	Accuracy	Sensitivity (aka Recall)	Precision	F1 Score
VGG16	0.5675	0.5675	0.6367	0.5552

From the table above, the VGG16 model did not perform well. The testing accuracy was only 56.75%. This is due to the fact that the model had difficulties with differentiating between brown and black hair. The same issue was found in differentiating between grey and white hair.

5. Experiments for Hair Top detection

The results shown below are for detecting thickness of the hair in an image. The matrix images with no cropping are used to detect hair top. The non-cropped images are used for detecting the thickness, because cropping can remove some of the hair in the face detection as well.

Table 5: Results produced from training the model for Hair Top using VGG16 architecture

CNN Architecture	Accuracy	Sensitivity (aka Recall)	Precision	F1 Score
VGG16	0.79	0.79	0.7580	0.7722

From the table above, the VGG16 model performed well with a testing accuracy of 79%. The accuracy for the model could have been higher, but it was difficult for the model to detect between bald, shaved or buzz cut in images. This is because they are all shaped similarly.

VI. Conclusion

In this paper, CNN was used to detect facial attributes such as wrinkles, freckles, glasses, hair color and hair top in images. VGG16 architecture was used as a branch architecture to create a model for each attribute. Five different model, which used the VGG16 architecture, were created to detect each facial attribute. All of the models had the same batch size, epoch, learning rate and optimizer used to train the model. The experiment produced good results for detecting all facial attributes, except for the hair color attribute, because it was harder for the model to differentiate between brown and black or grey and white. The model performed with really high accuracy of 99.5% for detecting freckles, but there were only around 18 images with freckles in the whole dataset. From the experiments, the conclusion can be drawn that the VGG16 architecture performed well at detecting facial attributers and with more dataset, it could have given bigger accuracy.

VII. Future Work

In the future, more dataset images should be used to train the model, which will increase the accuracy of the model. Also, different CNN architectures should be tested for creating a model to detect facial attributes. Equal image allocation should be done in the future for each attribute when training the model using the dataset images.

References

Basbrain, A.M., Al-Taie, I., Azeez, N., Gan, J.Q. and Clark, A., 2017, September. Shallow convolutional neural network for eyeglasses detection in facial images. In *2017 9th Computer Science and Electronic Engineering (CEEC)* (pp. 157-161). IEEE.

Brownlee, J., 2021. *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/#:~:text=Adam%20is%20a%20replacement%20optimization,sparse%20gradients%20on%20noisy%20problems.> [Accessed 13 April 2021].

Chin, C.L., Yang, Z.Y., Su, R.C. and Yang, C.S., 2018, September. A Facial Pore Aided Detection System Using CNN Deep Learning Algorithm. In *2018 9th International Conference on Awareness Science and Technology (iCAST)* (pp. 90-94). IEEE.

Chris Nicholson, C., 2021. *A Beginner's Guide to Convolutional Neural Networks (CNNs)*. [online] Pathmind. Available at: <https://wiki.pathmind.com/convolutional-network> [Accessed 13 April 2021].

Jiang, H. and Learned-Miller, E., 2017, May. Face detection with the faster R-CNN. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)* (pp. 650-657). IEEE.

N B, H., 2021. *Confusion Matrix, Accuracy, Precision, Recall, F1 Score*. [online] Medium. Available at: <https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd> [Accessed 13 April 2021].

Thomas, C., 2019. *An introduction to Convolutional Neural Networks*. [online] Medium. Available at: <https://towardsdatascience.com/an-introduction-to-convolutional-neural-networks-eb0b60b58fd7> [Accessed 13 April 2021].

Zhong, Y., Sullivan, J. and Li, H., 2016, June. Face attribute prediction using off-the-shelf CNN features. In *2016 International Conference on Biometrics (ICB)* (pp. 1-7). IEEE.