**ABSTRACT**

Nowadays, in computer vision experiments, facial image classification poses a basic issue. Image classification can be used to accelerate the training process, and considerably improve accuracy. In this paper, what is shown is how images are classified into three categories namely; Age, Ethnicity and Gender. This was done using the Deep-Convolutional Neural Network (CNN), and will be compared to another machine learning technique which will be discussed in the methodology section. The images used for this experiment are gotten from a data set from Kaggle called **age\_gender** containing about 27,000 images in total, grouped using the 3 variables mentioned earlier. In the end, the results show that the convolutional neural network that was built had a higher accuracy that the aforementioned second technique used, with the results being; https://github.com/DRoqeeb/Assessment.git

**INTRODUCTION**

Due to its complexity, facial recognition is one of the biometrics disciplines that has received the most research (Andrejevic & Selwyn, 2019). A feature could be the edges in an image, the pixel intensity, the change in pixel values, and many more. The human face has distinctive biological characteristics such as the skin colour, face shape, and facial attributes (wrinkles, hairstyles, beards) which contain a lot of biological information. Facial attributes vary amongst people with variables like age, gender, expressions, skin tone, etc. Face information allows humans to quickly evaluate face characteristics. The diverse features, such as numerous positions and occlusions in the natural world, make it an extremely difficult work for the computer. Identifying the gender of a face is a binary classification problem that typically involves the five steps of data collection, pre-processing, feature extraction, classification, and performance evaluation. Use of sensors (like cameras) and other auxiliary sensors is required for data collecting, which is quite tasking as a lot of data is required to be able to have enough samples for the experiment to be a success capable of being utilized in the real world.

Facial classification basically comprises identifying facial attributes and characteristics and, analysing and extracting them to compare with the already stored information in the database to then classify them using different classes. This aspect of computer vision can be applied in different aspects of real-life systems, such as in security, biometrics, robotics and even surveillance systems. Facial recognition and classification, which is the quick analysis and interpretation of lots of details about a face, for humans is nothing short of an easy task as there is little to no problem of doing so, whether it be an image of someone’s facial structure or someone that stands face-to-face with them. The same cannot be said for computers because a lot of work needs to be done for the system to even begin to recognize that the image or object of recognition has a face and then classify it. To perform this, a model for running the program is used in place of humans, by replacing human assistance with something that can mimic the inner workings of a human’s brain, which is called **Neural Network (NN)**.

For this project, what was used is the **Convolutional Neural Network (CNN)** model, which is a type of neural network. It is a machine learning technique that used images as its main resource for the process and stores various features that are extracted from the given images and uses them as references for future classifications of images that might be given to it. Recent studies of image classification have shown that using CNN has the model for the experiments has brought about more positive results than most other techniques [31 check the pdf].

**LITERATURE REVIEW**

There have been previous works done in this aspect of classification, some using the same techniques or at least similar ones to produce different results. In [Facial geometric feature extraction based emotional expression classification using machine learning algorithms], used multiple machine learning models, such as Decision Tree, Random Forest and Extreme Learning machine for their project of facial geometric feature extraction based on emotional expression classification. This helped produce a classification result of greater than 90 percent for all the training and tests of the data set.

One attempt at classification was made in [Animal classification using facial images with score-level fusion] where facial images were used to classify not humans, but animals. The proposed method was a score-level fusion was implemented, using CNN and other methods like Support Vector Machine (SVM) and Lagrangian Support Vector Machine (LSVM) among others, to perform the experiment. The result of this gave a very positive accuracy of over 95 percent for the final iteration. Another one, [Facial Skin Classiﬁcation Using Convolutional Neural Networks], made use of both the SVM and state-of-the-art Convolutional Neural Network techniques to assess and classify facial skin. It was mentioned that although neural networks have the downside of having to be intensively trained for them to function optimally, they are the most effective in classification tasks. The results of the experiment brought about an accuracy of 0.899, F-Measure of 0.852 and Matthews Correlation Coeﬃcient of 0.779.

[AgeandGenderClassiﬁcationusingConvolutionalNeuralNetworks] had almost the same project done as what was done in this report. The experiment in this one consists of using images gotten from Flickr, that were uploaded onto the Adience Benchmark, to test the already trained CNN’s accuracy. This led to a rating of 86.8 percent and 84.7 percent for gender and age classification respectively. [Design of a Face Recognition System based on Convolutional Neural Network (CNN)] aimed for the design and development of a face recognition system using Convolutional Neural Network to identify faces in a biometric system using grayscale images. The proposed CNN achieves a training accuracy of 99.78 percent and a validation accuracy of 98.7 percent.

[H. M. Moon, C. H. Seo, S. B. Pan, “A face recognition system based on convolution neural network using multiple distance face”, Soft Computing, Vol. 21, pp. 4995-5002, 2017] presented a long-distance face recognition method, solved by resolving the variation in recognition rate resulting from distance variation. A CNN was used for face recognition and the Euclidean distance was used to measure the similarity. The proposed method achieved excellent performance in various distances. [N. Ramanathan and R. Chellappa. Modeling age progression in young faces. In Proc. Conf. Comput. Vision Pattern Recognition, volume 1, pages 387–394. IEEE, 2006] uses a similar approach to model age progression in subjects under 18 years old. As those methods require accurate localization of facial features, a challenging problem by itself, they are unsuitable for in-the-wild images which one may expect to ﬁnd on social platforms. Kartali et al. [A. Kartali, M. Roglić, M. Barjaktarović, M.Đurić-Jovičić, M. M. Janković, Real-time Algorithms for Facial Emotion Recognition: A Comparison of Different Approaches, 2018 14th Symposium on Neural Networks and Applications (NEUREL), Belgrade, 2018, pp. 1-4] have reported the results of conventional (SVM, MultiLayer Perceptron (MLP)) and deep learning methods (CNN), Alexnet CNN, Affdex CNN) based facial expression recognition of four emotions (happiness, sadness, anger, and fear) and achieved maximum recognition accuracy of 85.05% using Affdex CNN.

**PROBLEM AND DATA SET DESCRIPTIONS**

**PROBLEM STATEMENT**

A complete facial classification system should be able to analyze, recognize and identify facial attributes which it would use to compare with the previously stored data in its memory for accurate classification into the right category. The problem this section is addressing is the classification of the images into the categories of age, ethnicity and gender with enough accuracy to consider it a ‘job well done’. These three categories are what make the difference between this experiment and all other already mention in **Section B**, as it would require more training to be able to predict and classify each image correctly. To solve this issue, the use of the machine learning method **Convolutional Neural Network (CNN)** was implemented as it is one of the best models one can use for image classifications for a data set. The significance of all this is to provide a means by which faces can be recognized with details to the facial attributes, for instance used in security cameras in places like airports, if by chance the airport staff are on the lookout for a person within a particular age range, a specific race and/or one gender or the other.

**DATA SET DESCRIPTION**

To train the proposed CNN for facial classification, we used a data set gotten from Kaggle, made by Nipun Arora [https://www.kaggle.com/datasets/nipunarora8/age-gender-and-ethnicity-face-data-csv]. The data set is a simplified version of a UTK data set the creator was working on that was made available to the public. This dataset includes 27305 rows and 5 columns and includes a CSV of facial images that are labelled on the basis of age, gender, and ethnicity. The images that are contained in the data set are stored in numerical format which are interpreted by the system when they are outputted to give out the actual images.

**METHODOLOGY**

**METHODS USED**

As already mentioned in various sections of this report, the machine learning technique which provided the solution to the classification issue of this experiment was the **Convolutional Neural Network** technique. [Introduction to Convolutional Neural Networks CNNs (8 December 2020), Juan Cruz Martinez] The CNN works by inserting an image into the system, along with some adjustable parameters (weights) based on the different objects of the image, and then classifying them into the required groups. One of the main capabilities of CNN is that it applies primitive methods for training its classifiers, which makes it good enough to learn the characteristics of the target object. The CNN algorithm is structured in a specific independent processing workflow which are as follows [Image Classification Using Convolutional Neural Networks: A step by step guide, [Devansh Sharma](https://www.analyticsvidhya.com/blog/author/devansh75/) — January 11, 2021]:

* Choosing a data set and inputting the image data set into the system
* Resizing the images to fit the network for optimal results
* Assigning labels and features
* Obtaining random samples by shuffling the data set
* Creating/Generating training data
* Splitting the data set for use in the CNN
* Training the CNN model
* Testing the model on the data set to check for accuracy

The above processing steps give a general idea of the workings of the CNN model’s ability to perform classification tasks. This is all done in various layers of the model, in the order of; the Convolution Layer, Pooling Layer, Activation Layer, Fully-connected Layer (where the Classification takes place). More details about this will be explained in the next sub-section.

Also, in this paper, another machine learning technique was used with the aim of it being a kind of reference for comparison for CNN to give a visual idea of its the effectiveness. That technique is the **Artificial Neural Network (ANN)**. The process of implementing this method is very similar to that of CNN, one main difference between the two is that even though both methods are basically the same, since CNN is technically a type of ANN, ANN requires less data to run than CNN does. ANN processes image data, text data and tabular data more easily.

**EXPERIMENTAL SETUP**

The main goal of this experiment is to recognize and classify faces using CNN as the primary model. Before anything else, here are some terms and keywords that one needs to understand:

* **Classification Categories:** As they have already been mentioned already mentioned multiple times in this paper, these are the groups in which the images are going to be classified/grouped by in this experiment.
* **Convolution Layer:** This is the core building block of the system and is the section that handles the processing of the inputted data (that is, the images from the data set). It is in this layer that the features of an image are obtained which are in turn used as the attributes for classification. In this layer, data or imaged is convolved using filters or kernels.
* **Kernels:** Image kernels are the parts of the convolution layer that are designed to help with extracting features from images to aid in the image classification (e.g. edge detection, sharpness, blur, etc). In the case where there are multiple convolution layers, the kernels in the early layers are only able to extract low-level features while the features that are extracted in the later layers are more complex and essential. The size of the kernels determines the size of the feature extracted. It is recommended for a network to use multiple convolution layers.
* **Filters:** Filters are small units that we apply across the data through a sliding window. The depth of the image is the same as the input, for a color image that RGB value of depth is 4, a filter of depth 4 would also be applied to it [Introduction to how CNNs Work by [Simran Bansari](https://medium.com/@simranbansari?source=post_page-----77e0e4cde99b--------------------------------), 2019]. The filters help the layer to store the features that have been extracted by the kernels on a feature map, which are then used as references to compare during classification.
* **Pooling Layer:** The pooling layer is one which helps to reduce the size of the feature map after convolution, to a size that can freely enter the GPU of the system, which reduces the computational complexity necessary to process the data. There are two types of pooling; **Max-pooling** is one that returns the maximum value from the portion of the image covered by the Pooling Kernel and the **Average Pooling** that averages the values covered by a Pooling Kernel. This layer commonly uses a 2x2 filter with a stride of 2 to perform this perform. Stride is the amount of features the sliding window skips along the width and height.
* **Activation Function:** This is a function used to increase the non-linearity of the network without affecting the receptive fields of convolution layers. Some common examples of this function are ReLu, Sigmoid, Softmax, Leaky ReLu, Softplus and Softsign. ReLu, Sigmoid and Softmax were the only three activation functions used in this project. Activation functions help classify the outputs of a network.
* **ReLu (Rectified Linear Unit):** This function is the most commonly used activation function in CNN [A. Krizhevsky, I. Sutskever, G. E. Hinton, “Imagenet classification with deep convolutional neural networks”, Advances in Neural Information Processing Systems, Vol. 25, No. 2, pp. 1097-1105, 2012] because it allows faster training of the data and produces better, more satisfactory results compared to other activation functions as past experiments in CNN have proven so.
* **Softmax:** This function helps to normalize all the probability between two or more variables obtained during a part of experiment. The sum of the resulting probability values will be always equal to 1. The formula is;

Probability 1 = (var1) / (var1 + var2 + var3)

Probability 2 = (var2) / (var1 + var2 + var3)

Probability 3 = (var3) / (var1 + var2 + var3)

* **Sigmoid:** Contrary to Softmax, it does not normalize any value which in turn does not always result in the sum of the variables being 1.
* **Dropout Function:** A dropout function is used to reduce the complexity of calculations and to enhance the performance of the CNN. The dropout technique reduces the number of connections and the number of neurons with weak connections resulting in reduced computation complexity. This helps reduce **overfitting**, which can be described as when a model synchronizes too much with training data and becomes less flexible to test on different data sets.
* **Fully-Connected Layer:** This is the layer in which the process of classification is performed. It has a function called **Flatten** in which the shape of the image data is usually specified. But for this CNN, as the function is used in the middle layer, the network will automatically figure it out. In this layer, the feature map of the image output is converted into a form of column or tabular structure and processed in the network to create a trained model ready to test for classification on a data set.
* **Dense Network:** This network layer is used to classify images based on the output gotten from the convolution layers. Each node in the network is connected through links to almost every other node in the network, which in turn helps a node to receive input from every node in the layer before its now. [Introduction to Convolutional Neural Network (CNN) using Tensorflow by [Govinda Dumane](https://medium.com/@govindadumane?source=post_page-----de73f69c5b83--------------------------------), 2020]

As already mentioned earlier, there are three classification categories in this experiment, which means that there will be three separate sections to run with each having some varying values such as; the number of epochs, dense networks and activation function. Also, because there are 3 categories, the number of convolution layers in each case could differ to produce optimal results. Each case has their convolution layers followed by a max-pooling layer and an activation layer, as well as a fully-connected layer with the Flatten function and a dense network. The size of images inputted into the network was fixed to 48x48x1, with 1 being the color of the images that was used for analysis, which is Grayscale.

**AGE**

The first category is the classification using Age, and it consists of 4 convolution layers, each having their own activation function, one dense network and a dropout function at the last two convolutional layers. In the first convolutional layer, the kernel size was set to 3x3 and the number of filters was 32, while the kernel size for the max-pooling layer was set to 2x2. The activation function chosen for use is the **ReLu.**

The second to fourth convolution layers have similar values with slight differences like the second and third layers having the number of filters changed to 64 from 32, while the third and fourth layers have dropout functions with values of 0.2 each. After which, there is the Flatten function for the fully-connected layer (classification layer) which has no value, followed by another dropout function with a value of 0.5, and finally a dense network at the end of the syntax with only one node and the activation function ReLu.

**ETHNICITY**

Next is the facial classification of the images using ethnicity. Similar to the age category, this one comprises a number of convolution layers, which is three in this case. All other attributes such as activation function, dense network and dropout function are present as well. In this category of classification, the first convolution layer also has its kernel size fixed to 3x3 but has a total of 16 filters, while the max-pooling layer has its kernel size has 2x2.

The second and third convolution layers are almost the same as the first with the sole difference being the increase in the number of filters from 16 to 32 and 64 respectively. All 3 layers made use of the activation function ReLu. Next is the classification layer, where the Flatten function resides with no value designated to it. Contrary to the age category, the ethnicity category consists of two dense networks, the first one containing 128 nodes using the activation function ReLu, and the second having 5 nodes and using the activation function **Softmax**.

**GENDER**

Finally, there is the gender category of classification, which is the last of the three. Very similar to the previous category, in this case there are also three convolution layers in total each with 3x3 kernel sizes and each having its own max-pulling layer with 2x2 kernel sizes. What differentiates this category from Ethnicity is the number of filters of the first and second convolution layers being 32 and 64 respectively, while the third has 64 filters.

Same as the other 2 classification categories, the activation functions used for these convolution layers are all ReLu. Afterwards, as usual the classification layer with the Flatten function, which also has no value designated to it, and the only dense network present with a single now and the activation function that was used being **Sigmoid**.