**Advanced Deep Learning and Computer Vision**

**Course-End Project**

**Name: Ryne Mascarenhas**

**Email id: rynem8@gmail.com**

**Performing Facial Recognition with Deep Learning**

1. **Project Details**

**Description**

**Objective:** Create a facial recognition tool using a relevant deep learning algorithm, leveraging the provided resources.

**Context:** You are working for Face2Gene, an American AI company that has developed a healthcare app for doctors. The app utilizes deep learning algorithms to aid in diagnosing patients for genetic disorders and their variants. It converts patient photos into de-identified mathematical facial descriptors, which are then compared to syndrome-specific computational-based classifiers to determine similarity. The app provides a prioritized list of syndromes with similar morphology and suggests phenotypic traits and genes for feature annotation and syndrome prioritization.

Management has given priority to empowering and entrusting the in-house AI team. As a new member of the team, your task is to build a baseline model for facial recognition. The goal is to further enhance the app's existing features and add more value to the business based on this baseline model.

**Dataset Details:** The ORL Database of Faces consists of 400 images from 40 different subjects. The images were captured at different times, under varying lighting conditions, with different facial expressions (open, closed eyes, smiling, not smiling), and with or without glasses. All the images have a dark homogeneous background, and the subjects are positioned upright and frontal with some tolerance for side movement. Each image has a size of 92x112 pixels and 256 grey levels per pixel.

**Data can be downloaded from the following link:**

<https://www.kaggle.com/datasets/kasikrit/att-database-of-faces>

1. **Code**

The following changes are done with respect to the dataset:

Since, the links of the dataset at <https://www.kaggle.com/datasets/kasikrit/att-database-of-faces> don’t work anymore and give the following error:

**Not Found**

The requested URL was not found on this server.

Therefore, the dataset provided in the reference material of LMS is used for the project. The dataset is provided in a single npz file which is explored for understanding the dataset. Unfortunately, the dataset only has 20 classes of image faces instead of the 40 in the original dataset.

1. **Results and Observations**

**Introduction**

This code demonstrates facial recognition using Convolutional Neural Networks (CNNs). It explores two approaches: training a CNN from scratch and using transfer learning with a pre-trained ResNet model.

**Part 1: Training without Transfer Learning**

**Data Loading and Preprocessing**

The code loads the ORL Faces dataset, which contains 400 images of 20 different subjects.

A screen shot of a computer

Description automatically generated

The images are resized to a uniform size and normalized to values between 0 and 1. Each of the unique person images is visualised as shown below.

A collage of different men's faces

Description automatically generated

**Model Architecture**

A CNN model is defined with the following architecture:

* Convolutional layer with 32 filters and a kernel size of (5, 5)
* Max pooling layer with a pool size of (2, 2)
* Convolutional layer with 64 filters and a kernel size of (5, 5)
* Max pooling layer with a pool size of (2, 2)
* Flatten layer
* Dense layer with 128 units and ReLU activation
* Dropout layer with a rate of 0.4
* Dense layer with 20 units and softmax activation

**A computer screen shot of a black screen

Description automatically generated**

The CNN model is defined as a sequence of layers. It consists of two convolutional layers, each followed by a max pooling layer. The convolutional layers use 32 and 64 filters, respectively, and the max pooling layers reduce the spatial dimensions of the feature maps by a factor of 2. The model also includes a flatten layer to convert the feature maps into a one-dimensional vector, a dense layer with 128 units and ReLU activation, a dropout layer with a rate of 0.4 to reduce overfitting, and a final dense layer with 20 units and softmax activation for classification.

**Training and Evaluation**

The model is trained on the training data for 100 epochs with a batch size of 32. The validation data is used to monitor the model's performance during training and to save the best model based on validation accuracy.

A screen shot of a computer

Description automatically generated A screenshot of a computer program

Description automatically generated

The trained model achieves a test accuracy of over 90%, demonstrating its ability to recognize faces effectively shown below.

A close-up of several men's faces

Description automatically generated

**Part 2: Training with Transfer Learning**

**Transfer Learning with ResNet**

The code utilizes transfer learning by using a pre-trained ResNet50V2 model as the base model. The base model is frozen, and additional trainable layers are added on top.

**Model Architecture**

The transfer learning model consists of:

* ResNet50V2 base model with frozen weights
* Rescaling layer to normalize the input images
* Global average pooling layer
* Dropout layer with a rate of 0.2
* Flatten layer
* Dense layer with 512 units and ReLU activation
* Dropout layer with a rate of 0.4
* Dense layer with 20 units and softmax activation

Since the reshaping of training and testing data for the transfer learning model is done to make it compatible with the pre-trained ResNet50V2 model. The ResNet50V2 model expects input images to have a shape of (224, 224, 3), where the first two dimensions represent the height and width of the image, and the third dimension represents the number of channels (RGB). However, the ORL Faces dataset contains grayscale images with a shape of (112, 92, 1). To make the dataset compatible with the ResNet50V2 model, the code reshapes the images to (112, 92, 3) and converts them to RGB format and visualised as shown below.

A collage of a person's face

Description automatically generated

By reshaping and converting the images to RGB format, the code makes the ORL Faces dataset compatible with the ResNet50V2 model, allowing it to be used for transfer learning. This doesn’t change the image itself but copies the same image grayscale in all 3 channels, which is why when viewed the images remain the same.

**Training and Evaluation**

The transfer learning model is trained on the training data for 200 epochs with a batch size of 32. The validation data is used to monitor the model's performance during training and to save the best model based on validation accuracy.

A screenshot of a computer

Description automatically generated A screen shot of a computer

Description automatically generated

The transfer learning model achieves a test accuracy of around 80%, which is lower than the model trained without transfer learning. This suggests that the pre-trained ResNet model that uses weights trained on coloured images is not well-suited for the task of facial recognition on grayscale images.

**Conclusion**

The code demonstrates the effectiveness of CNNs for facial recognition. The model trained without transfer learning achieves high accuracy, while the transfer learning model performs less well due to the mismatch in training image between the pre-trained model and the task at hand. Since there are hardly any open-sourced pre-trained models that are trained on the grayscale images it cant be said that the accuracy cant be improved by transfer learning.

**Additional Notes**

* The code uses TensorFlow and Keras for model training and evaluation and is provided separately as a PDF file.
* The ResNet50V2 model is a pre-trained model available in the Keras Applications library.
* The code is written in visual studio code and interpreted in a conda virtual environment. The virtual environment is built on python 3.10 and installed with TensorFlow 2.10 for parallel processing using a GPU.