**REPORT**

Submitted by,

NIDIN V NANDAN

MSc Computer Science With Data Analytics

Reg No: 223039

**TASK: CELEBRITY IMAGE CLASSIFICATION**

Dataset:

Different folders named on celebrity name containing their photos

CHOSEN MODEL-Convolutional Neural Network(CNN)

"

model = tf.keras.models.Sequential([

tf.keras.layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(128, 128, 3)), tf.keras.layers.MaxPooling2D((2, 2)),

tf.keras.layers.Flatten(),

tf.keras.layers.Dense(256, activation='relu'),

tf.keras.layers.Dropout(0.5),

tf.keras.layers.Dense(512, activation='relu'),

tf.keras.layers.Dense(5, activation='softmax')

])

model.compile(optimizer='adam',loss='sparse\_categorical\_crossentropy', metrics=['accuracy'])

"

Here

Input Layer: Accepts images of size 128x128 pixels with three color channels (RGB).

Convolutional Layers:

32 filters of size 3x3, using ReLU activation function.

Followed by max-pooling with a 2x2 window to reduce spatial dimensions.

Flattening Layer:

Flattens the output from the convolutional layers into a 1D array to feed into the densely connected layers.

Densely Connected Layers:

First dense layer with 256 neurons and ReLU activation.

Dropout layer with a rate of 0.5 to reduce overfitting.

Second dense layer with 512 neurons and ReLU activation.

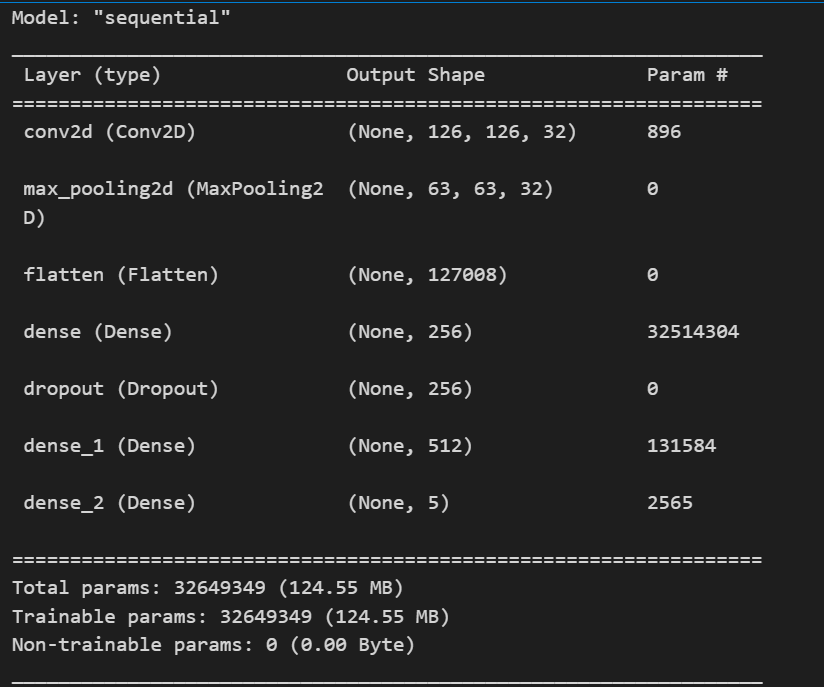
Final dense layer with 5 neurons, using the softmax activation function for multi-class classification (outputting probabilities for 5 classes).

Optimizer: Adam optimizer is used

Loss function: Sparse categorical cross-entropy, which is suitable for multi-class classification.

Metrics: Accuracy, to evaluate the model's performance during training.

**Model summary:**



**Training**:

history = model.fit(x\_train, y\_train, epochs=40, batch\_size=128, validation\_split=0.1)

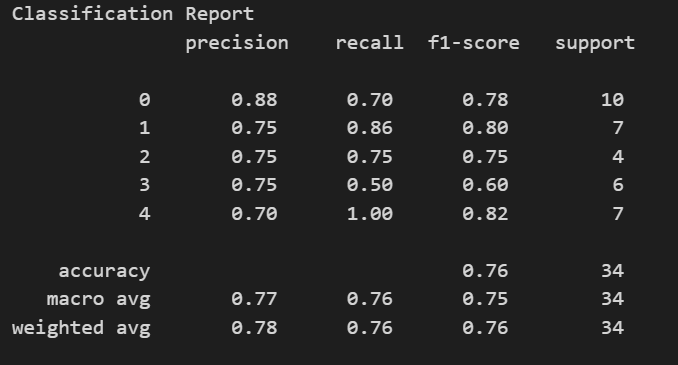
Here the model is trained for 40 epochs with batch size of 128

**Evaluation:**

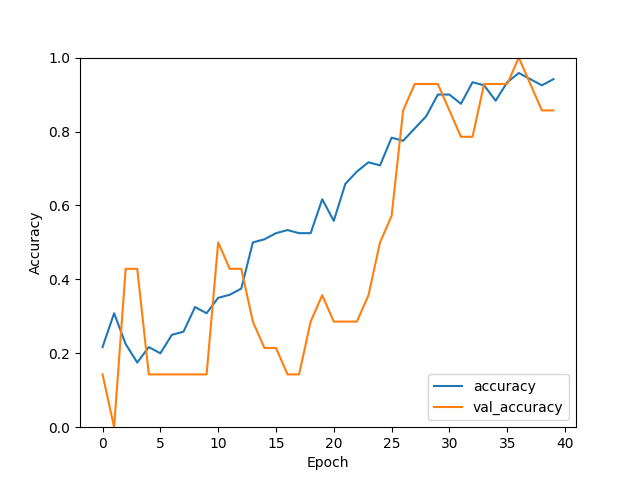
Here we evaluated the model on the test data and a classification report was generated

The accuracy was recorded as 76%

**Classification Report:**

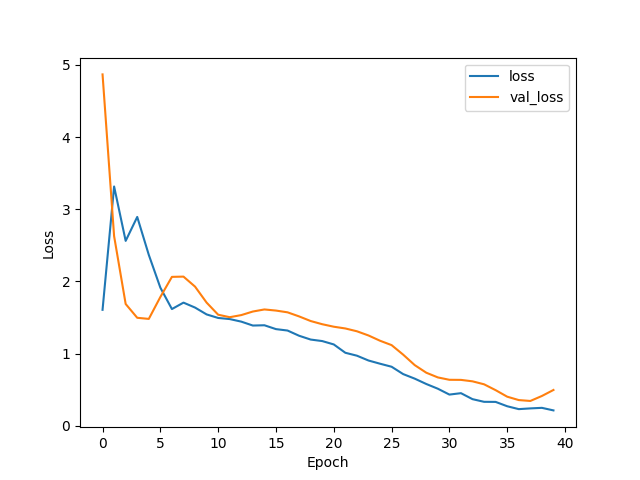


**Accuracy plot:**

****

During the training the accuracy and val\_accuracy is increasing for each epoch

**Loss plot:**

****

During the training period the loss is decreasing with increase in each epoch

**Prediction:**

A function "predict\_celebrity" was created in order to take the user input image a preprocessed for predicting the celebrity

And we can observe that the model is predicting correctly the celebrities among the inputted image