MODEL SUMMARY

The task is to develop an image classification model trained on the given celebrity image dataset. The model aims to classify any input image into one of five classes - Lionel Messi, Roger Federer, Maria Sharapova, Serena Williams, and Virat Kohli.

The model architecture used is Convolutional Neural Network (CNN) with a sequential model involving two convolutional layers. This model is mainly used to identify satellite images, process medical images, forecast time series and detect anomalies.

TRAINING PROCESS

Image Preprocessing:

- Load each image using OpenCV.
- Convert the image to RGB colour space.
- Resize the image to (128, 128).
- Convert the image to a NumPy array.

Model Architecture:

A sequential CNN model is used with the following layers,

- Convolutional layer with 32 filters, 3x3 kernel size, and ReLU activation.
- Max pooling layer with 2x2 pool size.
- Flatten layer to convert the 2D feature map to a 1D vector.
- Dense layer with 256 units and ReLU activation.
- Dropout layer with 0.1 dropout rate to prevent overfitting.
- Dense layer with 512 units and ReLU activation.
- Output layer with 5 units and softmax activation for multi-class classification.

Model Compilation:

Model is compiled using the adam optimizer along with sparse categorical cross-entropy loss function.

Model Training:

- Dataset is split for training using the train_test_split method from scikit learn in the ratio of 70% training and 30% testing data.
- Normalize the training and testing data using tensorflow to scale the pixel values between 0 and 1.
- Train the model for about 200 epochs with a validation split 0.1

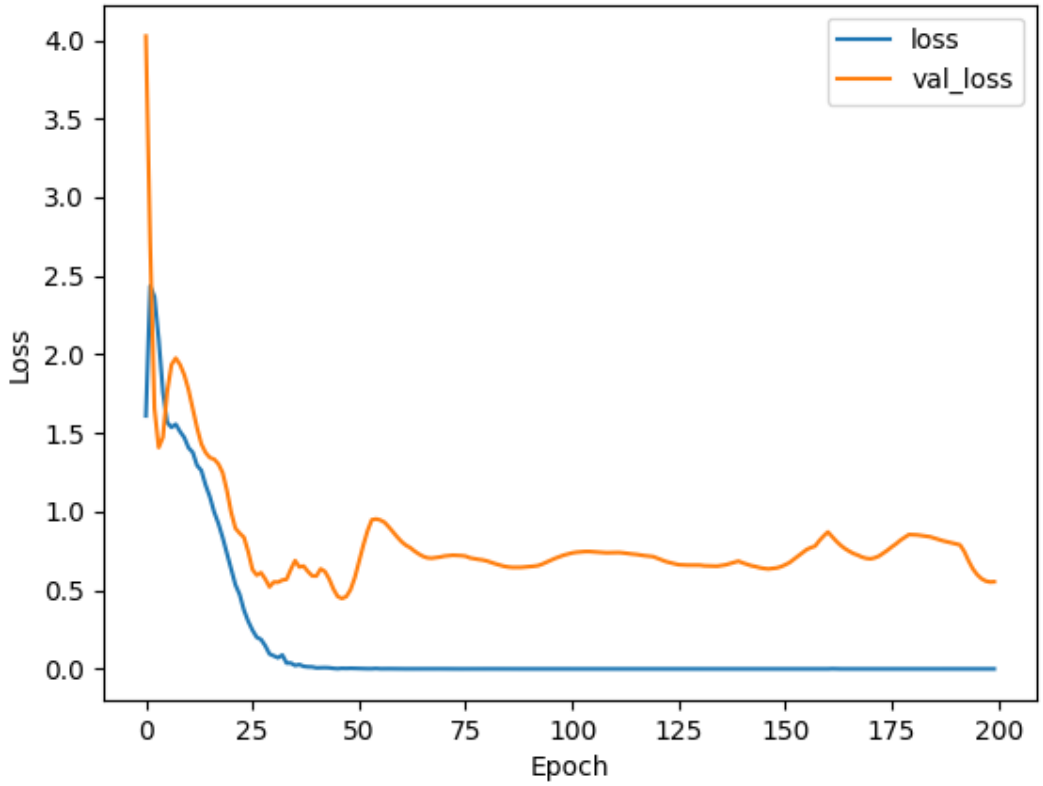
The model summary is as follows:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 126, 126, 32)      896

 max_pooling2d (MaxPooling2  (None, 63, 63, 32)        0
 D)

 flatten (Flatten)           (None, 127008)            0

 dense (Dense)               (None, 256)               32514304

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 512)               131584

 dense_2 (Dense)             (None, 5)                 2565

=================================================================
Total params: 32649349 (124.55 MB)
Trainable params: 32649349 (124.55 MB)
Non-trainable params: 0 (0.00 Byte)
```

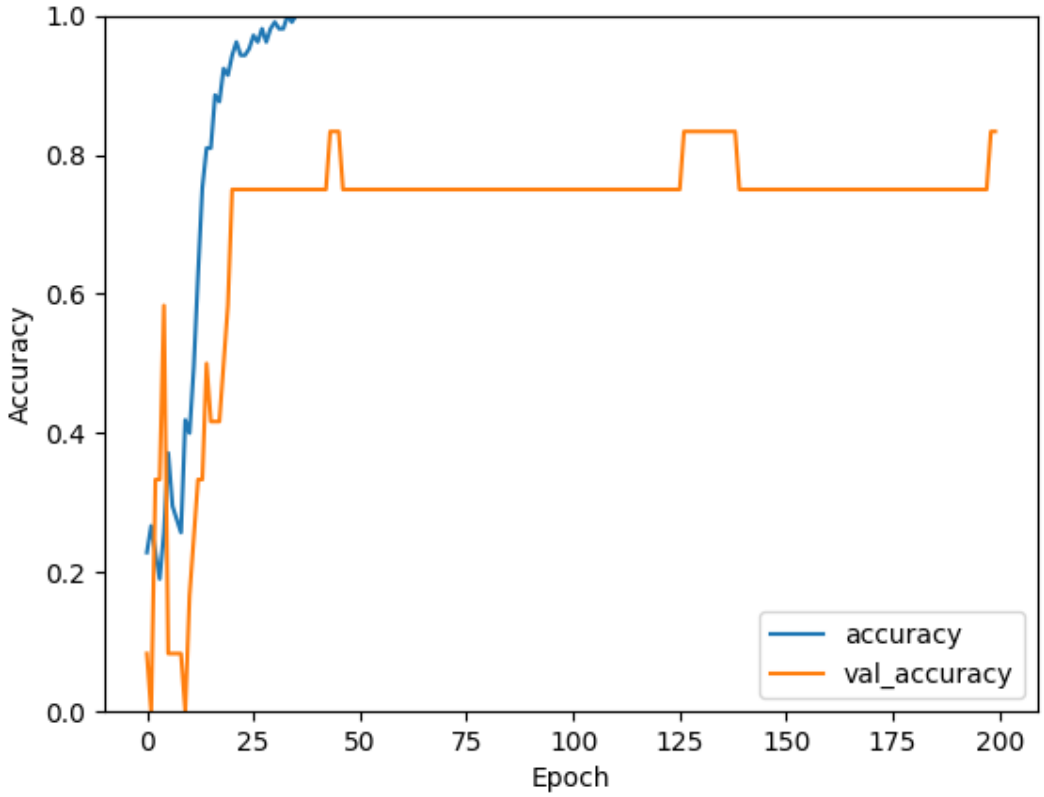
The resultant model has an accuracy of 84 %. The classification report obtained is as follows:

```
classification Report
              precision    recall  f1-score   support

           0       0.92      0.85      0.88        13
           1       1.00      0.80      0.89        10
           2       0.71      0.71      0.71         7
           3       0.78      0.78      0.78         9
           4       0.80      1.00      0.89        12

    accuracy                           0.84        51
   macro avg       0.84      0.83      0.83        51
weighted avg       0.85      0.84      0.84        51

--------------------------------------
```

The loss is plotted as follows:



The accuracy is plotted as given below:



CRITICAL FINDINGS

Despite achieving an impressive accuracy of 84% after 200 epochs, the model's performance is likely hindered by the limited size of the dataset, which consists of only 150 images. This small dataset makes it challenging for the model to learn generalizable features, leading to a tendency towards overfitting. To improve the model's performance, it would be helpful to increase the size and diversity of the training dataset, and to experiment with different regularization techniques.