**Image Classification**

For image classification, convolutional neural networks (CNNs) are the neural network architecture type that is recommended. To train the model, data augmentation is applied to the cropped image training set. The steps in the training process include building the model, fitting it to the training data, and evaluating it on the test and validation sets. The evaluation metric of choice is accuracy. The trained model is saved for later use, along with a class dictionary. An example image is used to show how to make predictions using the trained model.

• Data Preparation: Image files are gathered using the specified directory (path\_to\_data).

A DataFrame (df) is created with the file paths and the labels (person's name) that correspond to them.

Data sets used for testing, validation, and training are kept apart.

• Data Augmentation: ImageDataGenerator applies various techniques, such as horizontal flipping, rescaling, shearing, and zooming, to improve the data in the training set. The scales of the test and validation sets remain unchanged.

• Data Generators: The train\_generator, val\_generator, and test\_generator data generators are configured using flow\_from\_dataframe to load and preprocess images in batches.

• Model Architecture: A fully connected layer is positioned after several convolutional layers to extract features. For multi-class classification, a softmax activation layer is positioned after the fully connected layer.

• A sequential model is created using Keras.

Max-pooling convolutional layers are stacked to extract hierarchical features from images.

• Use the flattened layer to convert the 2D output to a 1D vector.

• Dense layers with dropout additions are added for classification.

• The number of neurons in the output layer with a softmax activation function equals the number of classes (individuals) in multi-class classification.

• Building the Model: To build the model, the Adam optimizer, the accuracy evaluation metric, and categorical cross-entropy loss—which is suitable for multi-class classification—are used.

• Model Training: The model is trained using validation data from the validation generator and ten epochs of training data.

• Model Evaluation: Using the test set, the trained model is evaluated to ascertain accuracy.

• Model Saving: The trained model is stored in a file named sports\_person\_model.h5.

A JSON file called class\_dictionary.json contains the mapping of labels to indices, or class dictionary.

• Creating Forecasts: sample\_image\_path, an example image path, is available.

The image is loaded, normalized, and then expanded to add a batch dimension.

The model predicts the class probabilities of the input image.

The predicted class index is obtained by finding the index with the highest probability.

The predicted class index is mapped to the person's name using the loaded class dictionary.

The person's name that is expected is printed.