MINI-PROJECT

Bird Species Classification using CNN

Done by

Anil Kumar Jujjuri

40020507

PROBLEM STATEMENT:

Birds play a huge role in life cycle. Many of the species are already extinct we know the less species where there are wide varieties of species, we can’t recognize them as many we cant see so by these project we can classify the bird which species it belongs to for these we need a trained model where we need to have a lot of data it takes much time to classify we can use the CNN to resolve these by using these we can train the model way faster and efficient and predict.

DATASET:

The Dataset is available publicly on the Kaggle website The classification goal is to predict the Bird species. The dataset provides the bird data information regarding to the images, Labels, Scientific names we are provided the images to the different things like train, test, valid in each it was consists around 400 classes in all. The train data consists around 58.3K and both test and validation consist of around 2K where in test and validation we consider the five images of each species.

Data Dictionary

Graphical user interface, text, application, email

Description automatically generated

Data source Links

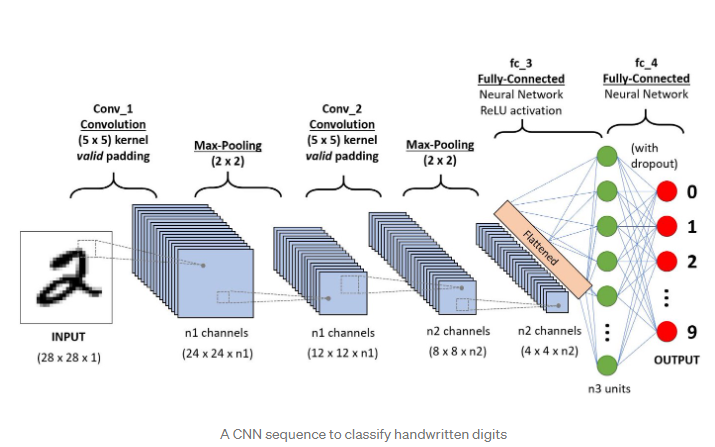
[BIRDS 400 - SPECIES IMAGE CLASSIFICATION | Kaggle](https://www.kaggle.com/datasets/gpiosenka/100-bird-species/code)

Task

To perform the CNN model for classification and predict the species the bird belongs to.

Algorithms/Model

A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlaps to cover the entire visual area.



Based on CNN we are using the MobileNetV3 and VGG19 to classify our image dataset

A convolutional neural network is a feed-forward neural network that is generally used to analyse visual images by processing data with grid-like topology. It’s also known as a ConvNet. A convolutional neural network is used to detect and classify objects in an image.

A CNN typically has three layers: a convolutional layer, a pooling layer, and a fully connected layer.

**Convolution Layer:**

The convolution layer is the core building block of the CNN. It carries the main portion of the network’s computational load. This layer performs a dot product between two matrices, where one matrix is the set of learnable parameters otherwise known as a kernel, and the other matrix is the restricted portion of the receptive field. The kernel is spatially smaller than an image but is more in-depth. This means that, if the image is composed of three (RGB) channels, the kernel height and width will be spatially small, but the depth extends up to all three channels.

**Pooling layer:**

Pooling layers are used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn, and the amount of computation performed in the network. The pooling layer summarizes the features present in a region of the feature map generated by a convolution layer.

Types of Pooling

**Max Pooling**

Average Pooling

Global max pooling

Global Average pooling

**Fully connected layer**

It takes the output of the previous layers, flattens them and turns them into a single vector that can be an input for the next stage.

**Dense layer** : The dense layer is a neural network layer that is connected deeply, which means each [neuron](https://machinelearningknowledge.ai/glossary/artificial-neuron/) in the dense layer receives input from all neurons of its previous layer. The dense layer is found to be the most used layer in the models.

**units** positive integer, it uses positive integer as it value and represents the **output size** of the layer.

**activation**: Activation function to use. If you don't specify anything, no activation is applied (ie. linear).

**Rectified linear unit(relu**): The **rectified linear activation function** is a linear function that will output the input directly if it is positive, otherwise, it will return output as zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

**SoftMax Activation function**: The SoftMax function is used as the activation function in the output layer of neural network models that predict a multinomial probability distribution.it is used as the activation function for multi-class classification problems where class is required on more than two class labels.

**Optimizer**: optimizers are algorithms or methods used to change the attributes of your neural network such as weights and learning rate in order to reduce the losses.

**Adam Optimizer**: Adam is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data.

**Early stopping:** Keras supports the early stopping of training via a callback called *Early Stopping*.

This callback allows you to specify the performance measure to monitor, the trigger, and once triggered, it will stop the training process.

The *Early Stopping* callback is configured when instantiated via arguments like monitor of Val accuracy, val\_loss, accuracy

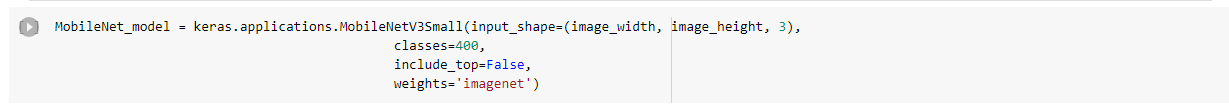
Experiment/Implementation

**MobileNetV3 small**

The core idea of **MobileNetV3 small** is a**convolutional neural network** that is tuned to mobile phone CPUs through a combination of hardware-aware network architecture search (NAS) complemented by the NetAdapt algorithm, and then subsequently improved through novel architecture advances.

Chart

Description automatically generated



Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

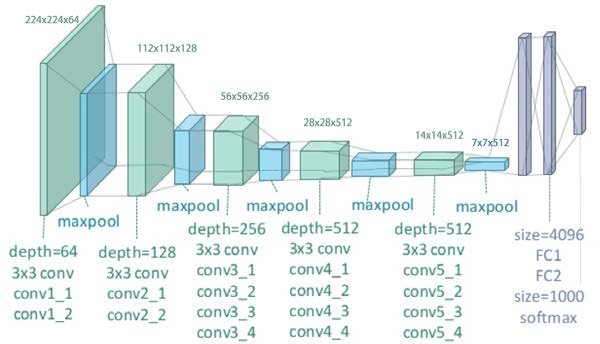
* In these model the MobileNetV3 can classify the images we are giving the weight as ImageNet where we can use the weight of the pretrained ImageNet weights.
* We are taking the keras.input is used to instantiate a Keras tensor.

A Keras tensor is a symbolic tensor-like object, which we augment with certain attributes that allow us to build a Keras model just by knowing the inputs and outputs of the model.

* We are taking the Global max pooling where we need to calculate the average of the whole matrix
* We are using the drop out which we helps us to resolve the over-fitting
* I have used the SoftMax which is used to get the output of the required label
* We are using the Adam optimizer which gives the faster and efficient results
* We are using the early stopping which helps the model to stop epochs where there is no improvement of learning rate

**VGG-19**

 VGG19 is an advanced CNN with pre-trained layers and a great understanding of what defines an image in terms of shape, colour, and structure. VGG19 is very deep and has been trained on millions of diverse images with complex classification tasks. I didn’t train VGG19 any further, only froze its layers and appended a shallow 2-layer network on top of it to perform my classification task of identifying species of birds.



* In these models the VGG19 can classify the images we are giving the weight as ImageNet where we can use the weight of the pretrained ImageNet weights.
* We are taking the keras.input is used to instantiate a Keras tensor.

A Keras tensor is a symbolic tensor-like object, which we augment with certain attributes that allow us to build a Keras model just by knowing the inputs and outputs of the model.

* We are taking the Global max pooling where we need to calculate the average of the whole matrix
* We are using the drop out which we helps us to resolve the over-fitting
* I have used the SoftMax which is used to get the output of the required label
* We are using the Adam optimizer which gives the faster and efficient results
* We are using the early stopping which helps the model to stop epochs where there is no improvement of learning rateGraphical user interface, text, application, email

  Description automatically generated

Experiments

By changing the Activation Function we can get these results for the MobileNetV3 model

Graphical user interface, application

Description automatically generated with medium confidenceGraphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

By Change the pooling layer as max pooling and Avg pooling we are getting error while getting the epochs

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Here we are changing the optimizer where we get the more epochs and it takes more time to train the model compared to sdg adam works more faster

Graphical user interface, text, application, email

Description automatically generated

By changing the activator as sigmoid there is change in the accuracy

Graphical user interface, text, application, email, Teams

Description automatically generated

Result

By these results we can know the accuracy of our trained model along with the predicted outputs

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

A picture containing text, bird, screenshot

Description automatically generatedGraphical user interface, text, application

Description automatically generatedA picture containing application

Description automatically generated

Conclusion:

Experiments are concluded for having different number of training set instances and set instances for bird species classification dataset.We have used the two models MobileNetV3 and VGG19 to classify our birds species the Vgg19 model by the accuracy we can say that MobileNetV3 is more suggested where In both the models we need not to give the layers manually the method itself having that data how the layers are classified In it. we need to take the hyperparameters like optimizers,activation function carefully which plays a key role in the training and accuracy of the model.Our model using MobileNetV3 predicts the bird species at high accuracy but In both the models the non-training parameters are more we need to make sure of that for the future scope.

References:

[tf.keras.applications.MobileNetV3Small  |  TensorFlow Core v2.9.1](https://www.tensorflow.org/api_docs/python/tf/keras/applications/MobileNetV3Small)

machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/

<https://iq.opengenus.org/vgg19-architecture/>

https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

<https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/>

[BIRDS 400 - SPECIES IMAGE CLASSIFICATION | Kaggle](https://www.kaggle.com/datasets/gpiosenka/100-bird-species)

https://keras.io/api/layers/core\_layers/input/