Task 7

Yahya Ramzy [02-SBIRfeatureX]

22 April 2023

1 ResNet50 CNN Model

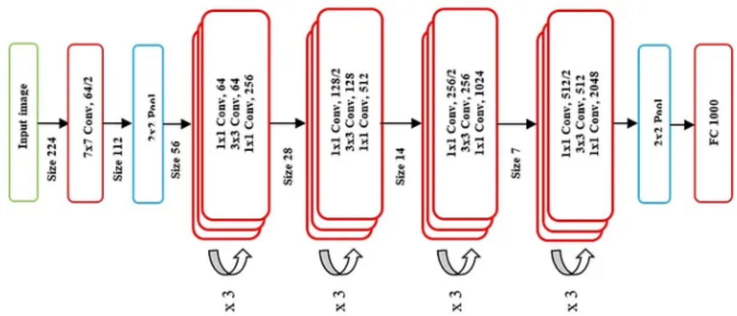

Figure 1: ResNet 50 Architecture

1.1 What is a ResNet-50 Model?

ResNet50 is a deep neural network architecture that has achieved state-of-the-art results in a variety of computer vision tasks. The name "ResNet" stands for Residual Network, which refers to the use of residual connections in the network. Residual connections enable the network to better propagate gradients through the network, allowing for deeper architectures to be trained effectively. ResNet50 is a 50-layer variant of the ResNet architecture and was introduced in the paper "Deep Residual Learning for Image Recognition" by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in 2016 [1]. Since its introduction, ResNet50 has become a popular model for image recognition tasks and has been used as a backbone in many state-of-the-art models.

1.2 Why ResNet-50 is popular?

ResNet50 is a popular choice for sketch recognition tasks because of its ability to learn rich and abstract features from images. Sketches, which are typically low-resolution and sparse, present a unique challenge for recognition algorithms. ResNet50's deep architecture allows it to learn complex representations of the input data, enabling it to effectively capture the important features of a sketch. Additionally, ResNet50 has been pre-trained on large datasets such as ImageNet, which has been shown to improve performance on downstream tasks like sketch recognition. Pre-training on a large dataset allows the network to learn general features that can be fine-tuned for specific tasks, like sketch recognition.

1.3 Training a ResNet-50 Model with our dataset.

1.3.1 Data Preprocessing

Before training the ResNet50 model, we preprocessed the dataset by resizing all images to 180x180 pixels and normalizing the pixel values to be in the range [0, 1]. We also applied data augmentation techniques to the training set, including random zooming, shifting, and shearing, to increase the size of the training set and reduce overfitting.

Method	Setting
Resize	180x180
Normalization	[(0,255)→(0,1)]
Zoom Range	0.15
Width Shift Range	0.2
Height Shift Range	0.2
Shear Range	0.15

Figure 2: Methods used in Data Preprocessing and their Settings for ResNet-50

1.3.2 ResNet-50 Model Architecture

ResNet50 is a deep neural network architecture consisting of 50 layers. The architecture is built upon the residual block, which allows for the efficient training of very deep networks. The residual block consists of two convolutional layers and a shortcut connection that bypasses the convolutional layers. The shortcut connection enables the gradients to flow more easily through the network, making it possible to train much deeper architectures. In addition to the residual blocks, ResNet50 includes downsampling layers that reduce the spatial dimension of the feature maps and increase the number of filters. These downsampling layers are responsible for extracting more abstract features from the input data. ResNet50 also includes global average pooling and a fully connected layer at the end of the network, which are used for classification. The ResNet50 model is loaded using the Keras library

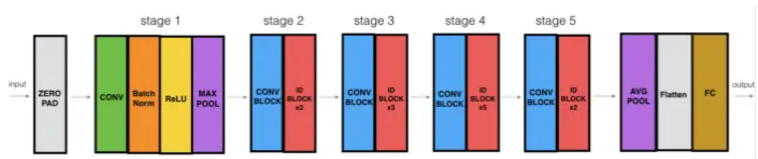

Figure 3: ResNet-50 Keras Model Architecture.

and its weights are initialized to pre-trained weights on the ImageNet dataset. The model is then set up to exclude the top layer, which is the fully connected layer that performs the classification on the ImageNet dataset. The input shape is set to 180x180x3, which is the size of the input image, and the pooling is set to global average pooling. The include top parameter is set to False to exclude the top layer of the pre-trained ResNet50 model. The classes parameter is set to 13 to indicate the number of output classes for the new classification task. All the layers in the pre-trained ResNet50 model are frozen by setting trainable to False. The ResNet50 model is then added to the Sequential model along with a Flatten layer to convert the output of the ResNet50 model to a 1D array. Two dense layers are added on top of the ResNet50 model to perform the classification task. The first dense layer has 512 units with the ReLU activation function, and the second dense layer has 13 units with the softmax activation function, which is used for multi-class classification.

Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 2048)	23,587,712
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 512)	1,049,088
dense_1 (Dense)	(None, 13)	6,669

Total params	Trainable params	Non-trainable params
24,643,469	1,055,757	23,587,712

Figure 4: ResNet-50 Keras Model Summary after freezing layers.

1.3.3 Transfer Learning

Transfer learning allows us to fine-tune a pre-trained model on a new task by reusing its learned features and weights. In this case, the ResNet50 model is pre-trained on the ImageNet dataset, which contains over a million images with 1,000 classes. The pre-trained model is then used as a feature extractor for the new task of classifying 13 different classes of images. By freezing the layers of the pre-trained model and adding new trainable layers on top, the ResNet50 model is adapted to the new task with much fewer parameters to train than if we started from scratch. Transfer learning not only saves computational resources and time but also improves the performance of the model, especially in cases where the target dataset is small

1.4 Training and Results

The code starts by compiling the ResNet50 model with the Adam optimizer with a learning rate of 0.001, using categorical cross-entropy as the loss function and accuracy as the evaluation metric. The fit generator function is then used to train the model on the specified number of epochs (in this case, 100) using the training and validation data generated by the train generator and test generator, respectively. The verbose parameter is set to 1, which means that progress bars will be displayed during training. Additionally, the EarlyStopping callback is used to stop the training process if the validation accuracy does not improve for 20 consecutive epochs.

The output of the training process is then displayed, showing the loss and accuracy metrics for each epoch on both the training and validation sets. As the training progresses, the loss decreases and the accuracy increases on both the training and validation sets. Eventually, the EarlyStopping criteria are met, and the training process is stopped early after 39 epochs. The final validation accuracy achieved by the model is 0.8577 and accuracy of 0.9615.

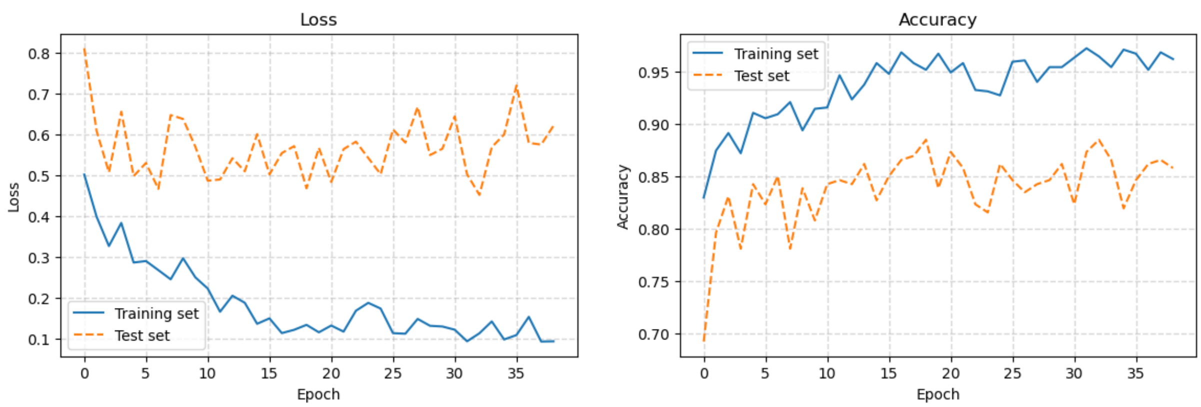

Figure 5: Loss and Accuracy Graphs of ResNet50 Model while training on the dataset.

1.5 Demo Exhibition 1

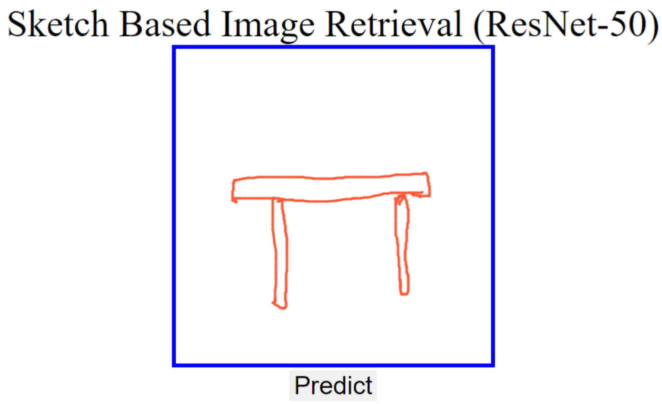

Figure 6: Input sketch of a table to the ResNet-50 Model.

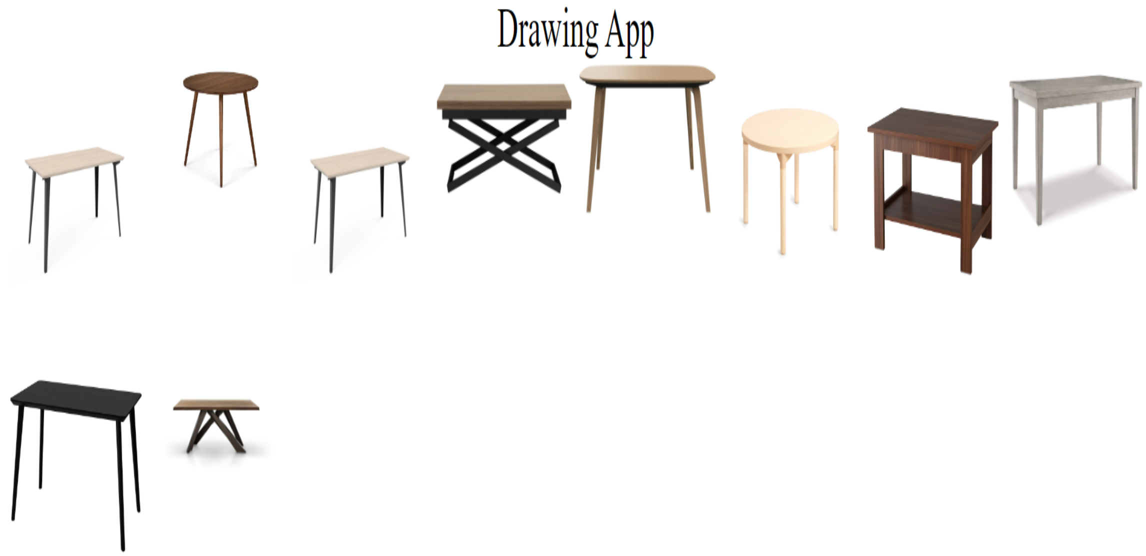

Figure 7: Results after the ResNet-50 Classified the sketch.

1.6 Demo Exhibition 2

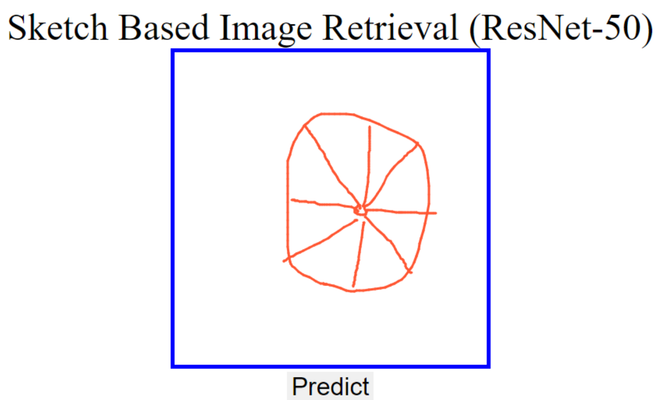

Figure 8: Input sketch of a wheel to the ResNet-50 Model.

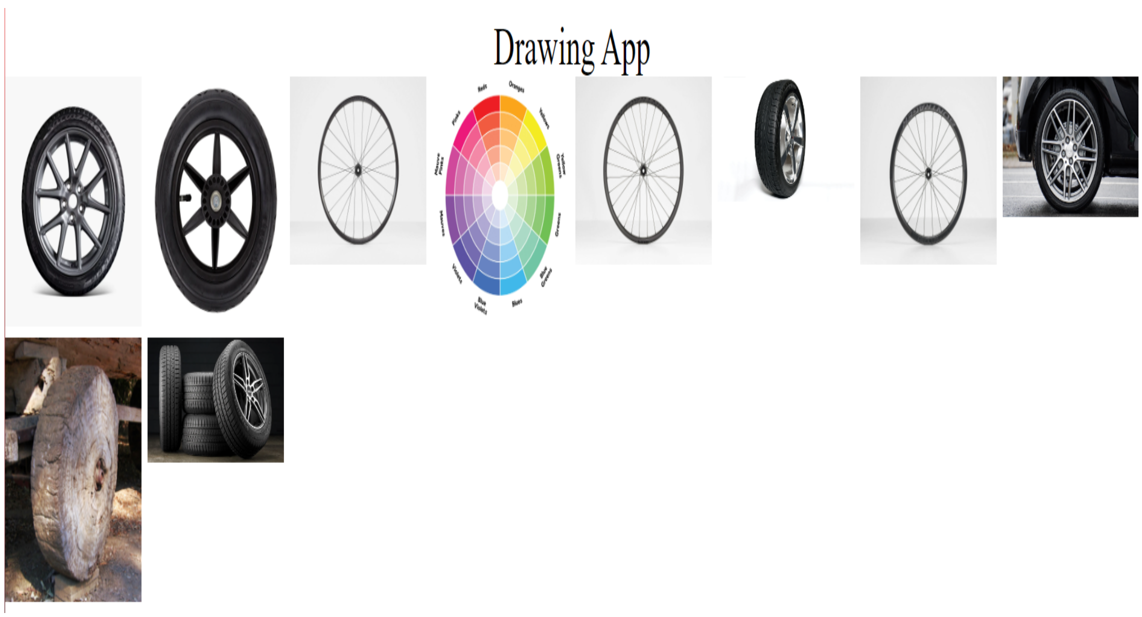

Figure 9: Results after the ResNet-50 Classified the sketch.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.