Task 8

Yahya Ramzy [02-SBIRfeatureX]

29 April 2023

1 Introduction

This thesis aims to compare the performance of traditional machine learning and transfer learning approaches in developing a system that can extract features from hand-drawn sketches and search for related images from a dataset based on the extracted features [1]. While traditional machine learning algorithms are used to analyze and identify features such as lines, shapes, and curves [2], transfer learning approaches can potentially improve the accuracy and efficiency of the system by leveraging pre-trained models for image recognition. The goal is to provide designers and artists with a streamlined tool for the creative process that can find related images based on their sketches. By comparing the performance of traditional machine learning and transfer learning approaches, this thesis aims to contribute to the field of design by identifying the most effective approach for developing such a tool. The technical challenges involved in designing feature extraction algorithms and machine learning models for image recognition are also discussed and evaluated. The potential impact of the developed system on various fields, including architecture, art, and engineering, is also explored.

2 Motivation

Sketch-based images are an essential part of the design process and a powerful means of representing visual information. However, extracting meaningful features from sketch-based images can be challenging due to their abstract and less detailed nature compared to other types of images. This poses a unique challenge when it comes to identifying and extracting features from such images. Despite these challenges, feature extraction from sketch-based images is an important area of research for several reasons. Firstly, it can help artists and designers find related sketches or images based on their own sketches, providing them with inspiration and guidance for their work. Secondly, it can aid in the creation of mood boards or design proposals, enabling designers to explore different variations and interpretations of their design. Finally, it can be used in educational settings, allowing students to learn about different design styles and techniques by searching for related sketches and images. By addressing these challenges and developing a system that can accurately and efficiently extract features from sketch-based images, this thesis seeks to contribute to the field of design and provide a new and innovative solution to a longstanding challenge.[3][4]

3 Problem Statement

Traditional machine learning algorithms have been used to address the challenges of feature extraction from hand-drawn sketches, but their accuracy and effectiveness are limited due to the abstract and less detailed nature of such images. Transfer learning approaches have shown promise in overcoming these limitations, and this thesis aims to compare the performance of traditional machine learning and transfer learning algorithms for feature extraction from hand-drawn sketches. The lack of well-defined edges and textures, as well as the complexity and variability of sketches, make it challenging to identify relevant features for further analysis and interpretation. By developing an applied system that can accurately and efficiently extract features from hand-drawn sketches, this research aims to overcome these challenges and enable more effective search and analysis of related images in a dataset. The successful development of such a system would have significant potential applications in fields such as design and education [1] [2].

4 Objectives

The objectives of this work are:

- Train various pre-trained models on sketch-based images to classify sketches.
- Compare the performance of pre-trained models with traditional machine learning approaches.
- Analyze the advantages and disadvantages of using pre-trained models versus traditional machine learning approaches.
- Evaluate the strengths and weaknesses of each pre-trained model trained.
- Demonstrate the potential applications of using pre-trained models and traditional machine learning approaches for sketch classification.

5 InceptionNetV3 CNN Model

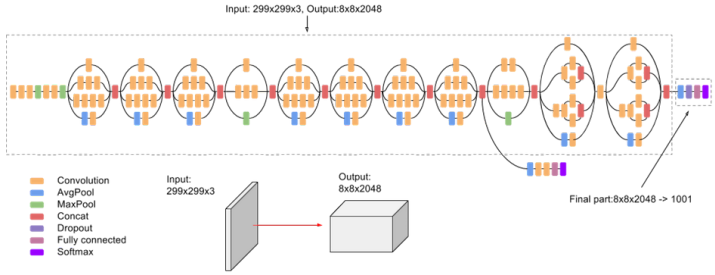

Figure 1: InceptionV3 Architecture

5.1 What is InceptionV3 ?

InceptionV3 is a deep convolutional neural network (CNN) architecture that was first introduced in 2015 by Google researchers as a part of the Inception family of models [**?**]. It is designed to achieve high accuracy in image classification tasks while minimizing the number of parameters in the model. InceptionV3 uses a combination of various convolutional layers, including 1x1, 3x3, and 5x5 convolutions, as well as pooling and inception modules that allow the network to efficiently process information at multiple scales. This architecture has been shown to outperform previous state-of-the-art models on the ImageNet dataset and has been widely adopted in many computer vision applications.

5.2 Why is InceptionV3 popular ?

InceptionV3 has gained popularity in the computer vision community due to its impressive performance on various image recognition tasks. It has been shown to outperform previous state-of-the-art models on the ImageNet dataset, which consists of over 1.2 million images with 1,000 object categories [**?**]. The model's success can be attributed to its unique architecture, which includes multiple convolutional layers with different kernel sizes and pooling operations, as well as the use of inception modules that allow for efficient feature extraction. Additionally, InceptionV3 has a relatively small number of parameters compared to other state-of-the-art models, making it more memory-efficient and easier to deploy on resource-constrained devices [**?**]. As a result, InceptionV3 has become a popular choice for image classification and object recognition tasks in a variety of applications, including self-driving cars, healthcare, and robotics.

5.3 Training InceptionV3 with our dataset.

5.3.1 Data Preprocessing

To train the InceptionV3 model, we used the ImageDataGenerator class in Keras to perform data preprocessing and augmentation. The train datagen object was defined with several image transformation parameters, including a rescaling factor of 1/255 to normalize the pixel values, a shear range of 0.4, a zoom range of 0.4, horizontal and vertical flipping, and a validation split of 0.2 to create a separate validation set. These transformations were applied to the training set images during training to increase the diversity and quantity of the training data, and to prevent overfitting of the model. Additionally, the validation split allowed us to monitor the performance of the model on unseen data during training, which helped us to adjust the model hyperparameters and prevent overfitting.

Method	Setting
Resize	224x224
Normalization	[(0,255)→(0,1)]
Zoom Range	0.4
Horizontal Flip	true
Vertical Flip	true
Shear Range	0.4

Figure 2: Methods used in Data Preprocessing and their Settings for Inception V3

5.3.2 InceptionV3 Model Architecture

The architecture is built upon the idea of "inception modules", which are convolutional layers with different filter sizes that are combined to capture features at different scales. InceptionV3 also includes down-sampling layers that reduce the spatial dimension of the feature maps and increase the number of filters, and global average pooling is used for feature aggregation. The InceptionV3 model is loaded using the Keras library with pre-trained weights on the ImageNet dataset. The top layer of the pre-trained InceptionV3 model is excluded by setting include top to False, and the input shape is set to 224x224x3, which is the size of the input image. All layers in the pre-trained InceptionV3 model are frozen by setting trainable to False. The InceptionV3 model is then added to a new model using functional API along with a GlobalAveragePooling2D layer, a dense layer with 1024 units and ReLU activation, a dropout layer with 0.5 rate, and a dense layer with 13 units and softmax activation, which is used for multi-class classification.

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
inception_v3 (Functional)	(None, 5, 5, 2048)	21802784
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0
dense (Dense)	(None, 1024)	2098176
dropout (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 13)	13325

Total params	Trainable params	Non-trainable params
23,914,285	2,111,501	21,802,784

Figure 3: Inception V3 Keras Model Summary after freezing layers.

5.4 Training and Results

In order to train the InceptionV3 model for the classification task, we need to compile the model with an optimizer, loss function, and evaluation metric. In this case, we use the Adam optimizer with a learning rate of 0.0001, categorical cross-entropy loss function, and categorical accuracy as the evaluation metric. To prevent overfitting and improve the generalization of the model, we use early stopping and learning rate reduction callbacks during the training process. The early stopping callback monitors the validation loss and stops the training process if there is no improvement for 5 consecutive epochs, while the reduce learning rate on plateau callback reduces the learning rate by a factor of 0.1 if the validation loss does not improve for 2 consecutive epochs. The model is trained for 25 epochs with a batch size of 32. The fit method is called on the InceptionV3 model with the training and validation data generators, the early stopping and reduce learning rate callbacks, and the steps per epoch and validation steps specified to ensure that all the samples are processed during each epoch. The results of the training show that the model's loss and accuracy improved over each epoch. The validation accuracy reached a peak of 0.8594 in epoch 11 and then started to fluctuate. However, the model's accuracy and loss stabilized after epoch 13. At the end of the 25 epochs, the model achieved an accuracy of 0.8731 on the test set, with a loss of 0.4274. The use of callbacks helped to prevent overfitting and allowed the model to achieve its best performance without sacrificing generalization.

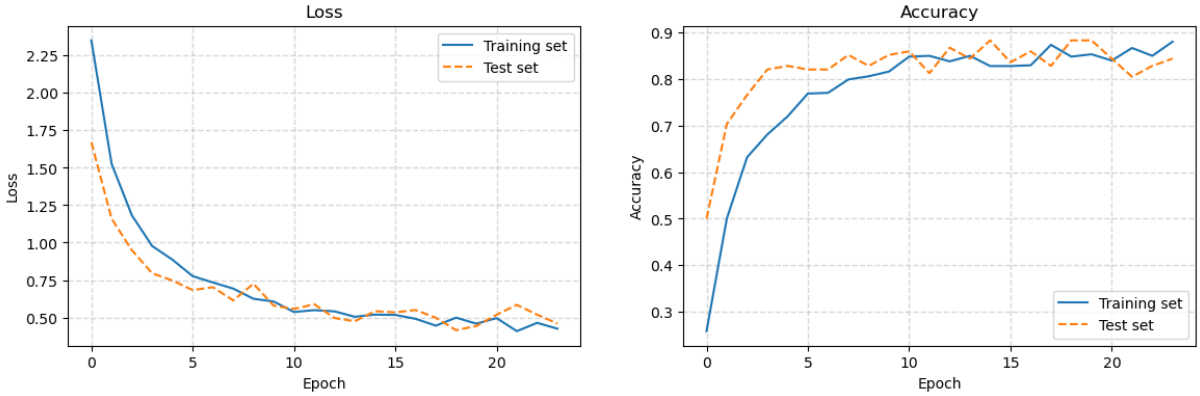

Figure 4: Loss and Accuracy Graphs of InceptionV3 Model while training on the dataset.

5.5 Demo Exhibition 1

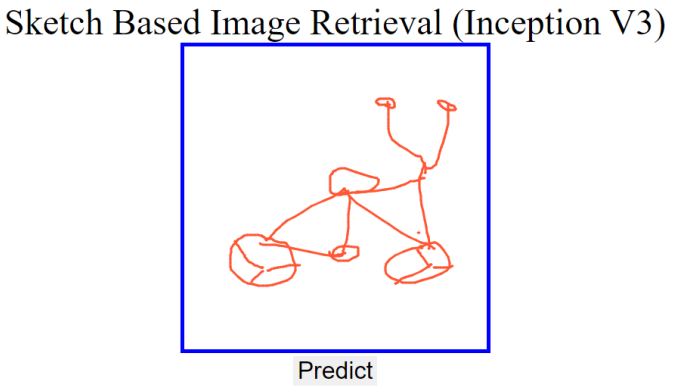

Figure 5: Input sketch of a bicycle to the InceptionV3 Model.

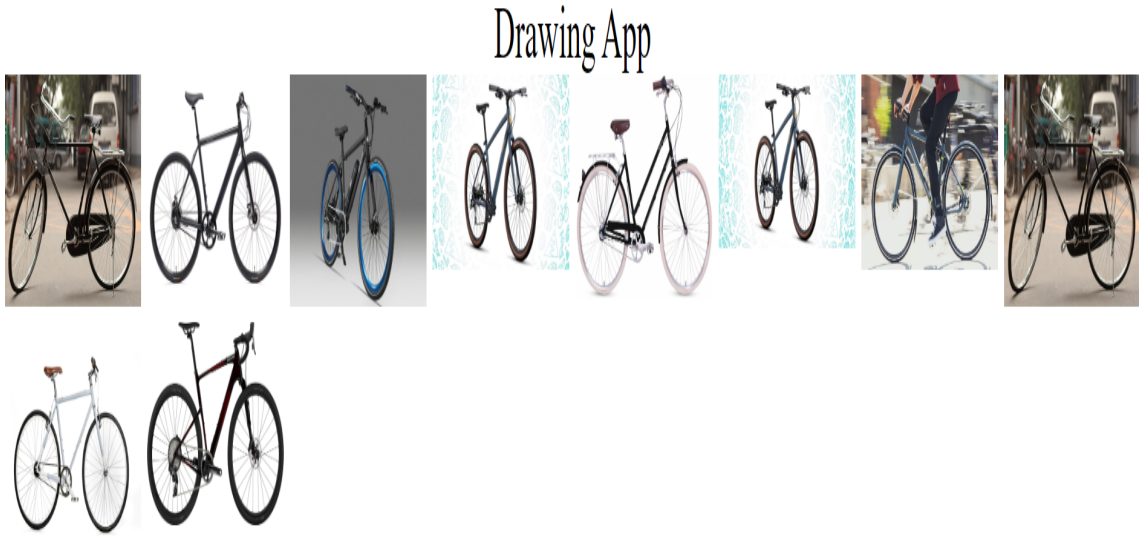

Figure 6: Results after the InceptionV3 Classified the sketch.

5.6 Demo Exhibition 2

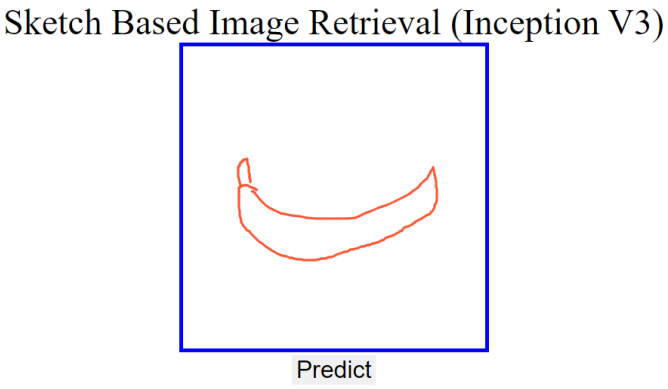

Figure 7: Input sketch of a banana to the InceptionV3 Model.

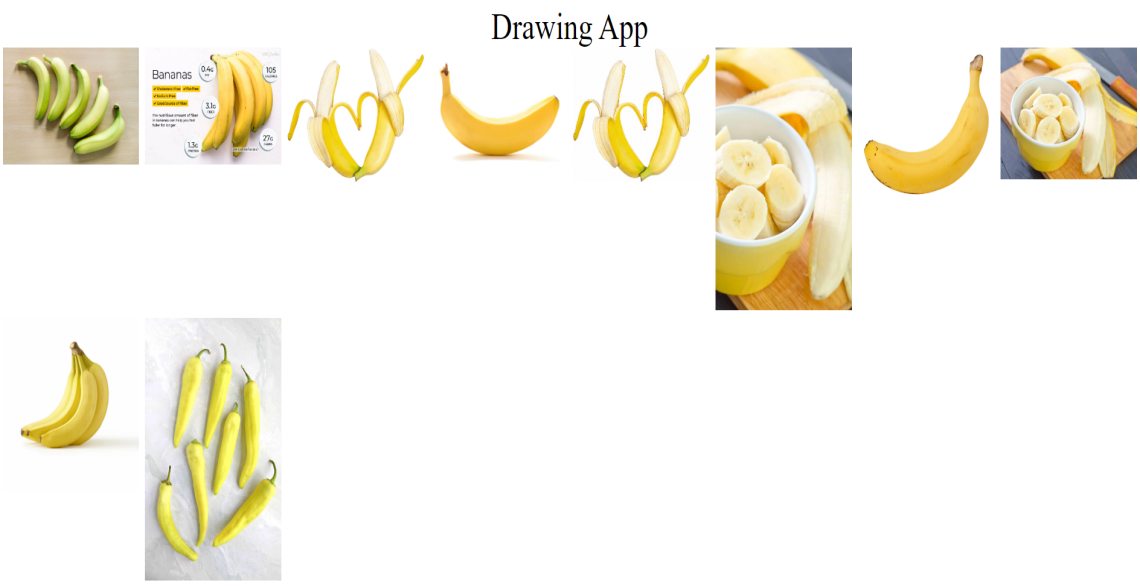

Figure 8: Results after the InceptionV3 Classified the sketch.

References

[1] N. R. Wilburn and R. R. Martin, "Image retrieval using sketches and features extracted from sketches," in *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pp. 173–180, ACM, 2004.

[2] Y. Wang, C. Wang, and L. Zhang, "Deep sketch hashing: Fast free-hand sketch retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 370–378, 2016.

[3] S. Chen, S. Liu, Y. Li, L. Li, H. Zhang, and H. Lu, "Sketch-based 3d shape retrieval using convolutional neural networks," *Computers & Graphics*, vol. 67, pp. 12–21, 2017.

[4] S. Liu, L. Li, and H. Lu, "Sketchyscene: Richly-annotated scene sketches," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 918–927, 2019.