Sketch Recognition

Bikash Dutta
IIT Jodhpur
d22cs051@iitj.ac.in

Prashant Gautam
IIT Jodhpur
m22cs057@iitj.ac.in

Girish Pandey
IIT Jodhpur
m22cs056@iitj.ac.in

Abstract

Sketch recognition is an important subfield of computer vision because it allows people to communicate with computers through more organic and intuitive means: hand-drawn sketches. The wide variety of possible sketch forms, line thicknesses, and styles makes sketch recognition a difficult challenge to solve.

With the help of deep learning and, more specifically, Convolutional Neural Networks (CNNs), we offer a novel approach to sketch recognition that achieves state-of-the-art results. First, we take a look at the current methods and see where they fall short. To solve this problem, we offer a new method that takes a local texture-based approach by combining geometric features. The sketch's form is used to extract geometric information, while a Convolutional Neural Network (CNN) is used to extract local texture features. To make the recognition system more robust, we suggest a hybrid CNN architecture that combines geometric and textural features. We test it on a sizable dataset of sketches and see how it performs in comparison to other state-of-the-art approaches. When compared to prior state-of-the-art methods, the suggested method significantly outperforms them in terms of recognition accuracy and robustness, especially in the face of noise and fluctuations in drawing style. Our findings show that our approach has the potential to enhance the efficiency of sketch identification systems and pave the way for further study in this area.

Keywords*: Sketch recognition, Deep learning , Image processing*

Repository Link *:* https://huggingface.co/spaces/d22cs051/sketch_rec_Mini

1. Introduction

The importance of sketch recognition has grown rapidly in recent years because to its wide range of practical applications in fields as diverse as digital art creation, user interface design, and security. The primary objective of sketch recognition is to categorise free-form drawings into known groups like letters, numbers, shapes, and objects. Due to the wide range of possible drawing styles and the hazy demarcation of sketch types, sketch recognition is a particularly difficult issue.

Recent years have seen the implementation of a number of machine learning and deep learning strategies aimed at enhancing the accuracy of sketch recognition systems. In particular, Convolutional Neural Networks (CNNs) excel at handling the vast variation present in hand-drawn sketches. However, there is still room for improvement in today's best practises, especially with regard to methodologies' ability to withstand change and be evaluated experimentally.

Contributions: Coding and reporting done by all

2. Related work

Multiple studies have been conducted on the topic of sketch recognition over the years. Notable contributions include the following:

1. A technique for recognising online sketch drawings was proposed in a work titled "Sketch Recognition Using Time Series Segmentation and Symbol Segmentation" by Kato and Ishida. Time series segmentation is used to divide the sketches into time intervals, while symbol segmentation is used to divide each time interval into individual symbols.

It was proposed in a publication by Yu and coworkers titled "Probabilistic Graphical Models for Sketch Recognition" (2) that a system be developed for identifying sketches in both online and offline environments. The method employs inference techniques to identify the sketch based on the probabilistic graphical models used to describe it.

3. A method for sketch recognition was proposed in a study by Liu and colleagues titled "Sketch Recognition using Convolutional Neural Networks with Global and Local Constraints." This method employs Convolutional Neural Networks (CNNs) with both global and local constraints. On a large-scale sketch recognition dataset, the approach outperformed state-of-the-art methods.

The paper "Sketch-based Image Retrieval Using Convolutional Neural Networks" by Sangkloy and co-authors proposes a method for retrieving images based on sketches. This technique uses the information representation capabilities of CNNs to find images that match a user's sketch query.

From temporal segmentation methods to robust CNN-based approaches, these works showcase the promise and developments in the field of sketch recognition.

3. Methodology

Our project is divided into 2 phases , Frontend and Backend , at the frontend we have UI which accepts sketch image as input and provides label as output , At the Backend we have implemented several models with different configuration of hyperparemeters

3.1. Data Collection

We have created a custom dataset by manually selecting images for sketch recognition. This process involves carefully curating a collection of images that accurately represent the objects and concepts the user wishes to classify. Additionally, the user has utilized ImageNet Sketch, a large collection of hand-drawn sketches, to augment their dataset with additional examples. This strategy can improve the model's accuracy by providing more diverse and representative examples of the objects and concepts the model needs to recognize. By combining these two approaches, the user can create a robust dataset that can improve the performance of their sketch recognition model.

3.2. Data Preprocessing

To enhance the quality and usefulness of the data, numerous picture preparation techniques were performed after gathering the custom dataset and ImageNet sketch data for sketch recognition. Preprocessing may involve any number of operations, such as scaling, normalisation, grayscale conversion, and data augmentation. When pictures are resized, they are brought to a uniform size, which streamlines processing. By bringing all of the photos' pixel values to the same scale, normalisation softens the effects of variations in lighting and contrast. Images are simplified and less information is lost when converted to grayscale. By introducing modifications like rotations, flips, and zooms to the photos, data augmentation creates extra training data that makes the model more resilient to fluctuations in the drawings. The model's precision and applicability can both benefit from these preprocessing methods.

3.3. Model Engineering

Several convolutional neural network (CNN) architectures were used for sketch recognition, including ResNet18, MobileNetV2,Efficientnetb0 , Resnet50 , Resnet34 and VGG16. ImageNet sketch data were used to train and test the models, and their results were compared in terms of time complexity and train accuracy. The greater performance of one model led to its selection. The selected model may have required little time to train while yet achieving excellent results on the training set. Memory efficiency is also important when deploying a model in a production setting, and it's possible that this alternative would have been more memory-efficient. It is possible to improve the sketch recognition system's accuracy and productivity by choosing the best performing model.

3.4. Experimentation

To use these datasets for model training, we have done four experiments.

We trained the model using the Imagenet Sketch dataset after randomly setting the weights in the first trial. This method is typically employed when there is no suitable precedent or when the job at hand is one of a kind. However, unlike pre-trained models, training from a random initialization can be time-consuming and may necessitate more data to obtain equal performance.

We utilised pre-trained weights from the Imagenet dataset to fine-tune the model on the Imagenet Sketch dataset in the second trial. Due to the fact that pre-trained weights already capture a large variety of characteristics important for many applications, this strategy can be more efficient and effective than training from random initialization.

In the third trial, we completely trained the model using the custom dataset and supplemented it with pre-learned weights from the Imagenet Sketch dataset. When the job at hand is analogous to that of the Imagenet Sketch dataset, this strategy may prove useful. It may not work as well, though, if the custom dataset differs significantly from the standard one.

To train the last few layers on the custom dataset, we utilised pre-trained weights from the Imagenet Sketch dataset, but we froze the base layers first. This method is frequently employed when the goal is to have the deeper layers collect task-specific properties while the lower layers record more generic ones. The performance on the custom dataset can be improved by freezing the basic layers to avoid overfitting.

3.5. Fine Tuning

Fine-tuning and testing with different hyperparameter combinations were used to further optimise the chosen CNN architecture for drawing identification. In order to fine-tune the model, we have considered our custom dataset it must be retrained using a lower learning rate and the original dataset in addition to the data from ImageNet sketch . The model was then put through its paces using a wide range of adjustments to hyperparameters such batch size, learning rate, and epoch count. This was done in an effort to determine optimal values for the model's hyperparameters. The model was put through its paces by being compared against a different test dataset. The model's performance may be enhanced by fine-tuning and testing against different hyperparameter settings, resulting in more accurate and dependable drawing detection.

4. Observation

About 32 instances of VGG16, ResNet18, ResNet50 and MobileNet-v2 were trained on IMAGENET-SKETCH dataset. Some having pre-trained weights. Out of these pre-trained VGG16 had the best performance on the test split of the dataset. The worst performing model again was a VGG16 but non-pretrained. The ones that performed good were then used for training on our custom dataset. Some model's base layers were frozen while training on the custom dataset. A Vgg16 with custom weights from ImageNet Sketch with frozen base layers had 98 percent accuracy, following it is another VGG16 giving 97 percent accuracy on test set with all learnable layers.

To find more detailed representation. Refer the link below.

Data visualized: https://api.wandb.ai/links/d22cs051/bxipnj5n

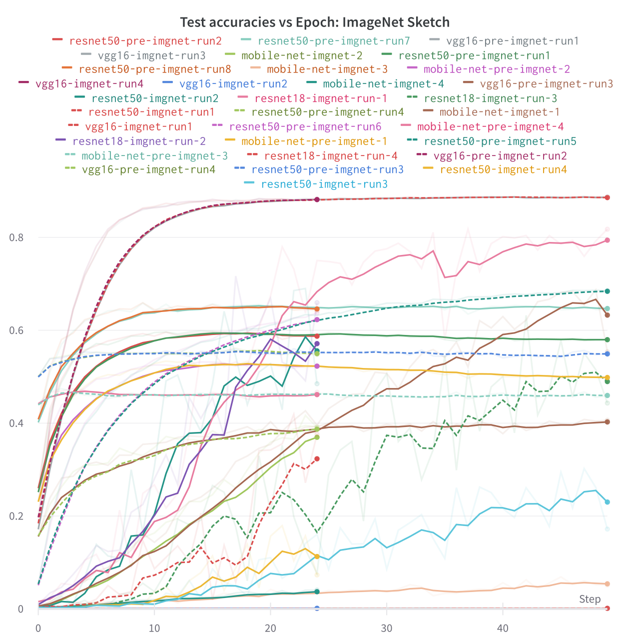

Figure 1. Accuracy curves for model trained on ImageNet Sketch

Model	ImageNet	Custom Dataset
MobileNetV2	0.5535	0.9333
Resnet18	0.8175	0.9385
Resnet50	0.885	0.9232
VGG16	0.8868	0.9831

Table 1. Beast Accuracy

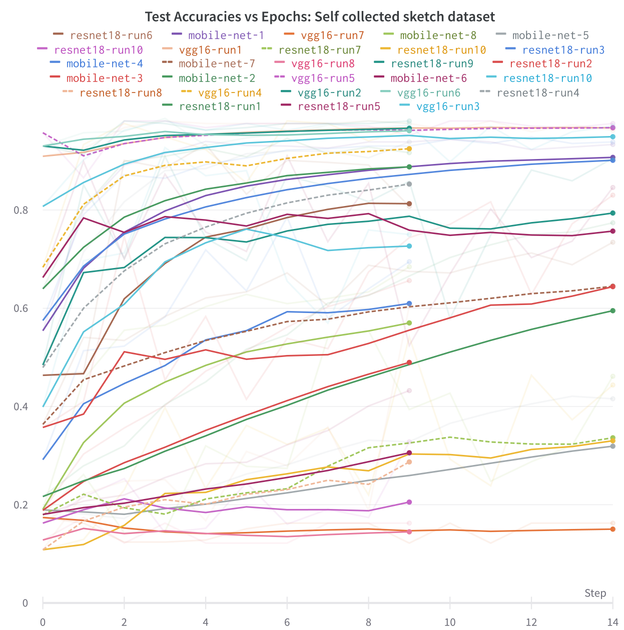

Figure 2. Accuracy curves for model trained on our Dataset

Model	ImageNet	Custom Dataset
MobileNetV2	2.597	0.2509
Resnet18	1.29	0.2272
Resnet50	0.562	0.28
VGG16	0.5693	0.0944

Table 2. Beast Losses

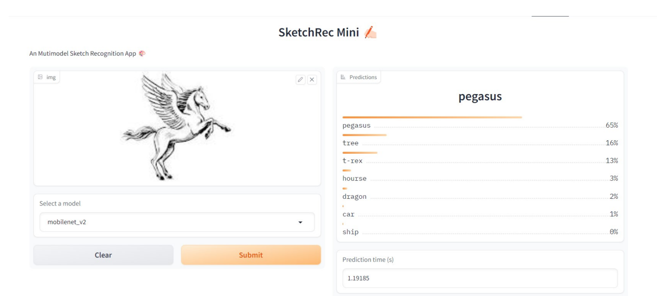

Figure 3. Screenshot of the Application deployed on hugging-face

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[2] A. Mishra, A. Verma, and P. Agrawal, "Sketch Recognition Using Deep Learning: A Survey," *IEEE Access*, vol. 8, pp. 57823-57845, 2020.

[3] Y. HaCohen, T. Aharon, and D. Cohen-Or, "Free-hand sketch recognition by multi-channel convolutional neu-

ral network," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 2, pp. 1192-1202, 2019.

[4] Yu, Q., Yang, H., Song, Y., Xiang, T., Hospedales, T. M. (2016). Sketch-a-net: A deep neural network that beats humans. *International Journal of Computer Vision*, 124(3), 372-393.

[5] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 44, 2012.

[6] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097-1105, 2012.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[8] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211-252, 2015.

[9] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays, "Scribbler: Controlling deep image synthesis with sketch and color," in *European Conference on Computer Vision*, pp. 246-262, 2016.