Lab Project II

“A Comparative Study of CNN and Vision Transformer in Sketch Recognition”

Submitted By-

Amrin AKTER |Hajar TOUBALI

Trainers

Phlypo Ronald |Guyader Nathalie |Dohen Marion

DALETh

Grenoble INP - Phelma, UGA

Contents

1 Introduction

In the domain of computer vision, considerable attention has been devoted to object recognition tasks primarily involving images in black-and-white, grayscale, or color spaces. However, a notable gap exists in the literature regarding the recognition of objects from human sketch drawings. Sketch recognition poses unique challenges due to the abstract and simplified nature of sketches, raising questions about the applicability of existing models trained on traditional images.

Generally, Convolutional Neural Networks (CNNs) have been the go-to choice for image recognition tasks, demonstrating remarkable performance across various domains. Architectures like ResNet-18 and AlexNet have been widely adopted and optimized for tasks involving standard images. While some research has delved into their ability to recognize objects from human sketch drawings, the extent of exploration in this area remains insufficient.

Similarly, emerging approaches like Vision Transformers (ViTs) have shown promise in capturing global dependencies within images, albeit in contexts primarily focused on traditional image datasets. The question arises whether these models, designed and trained on image datasets, can generalize well to the domain of sketch recognition, or if specific adaptations or tuning are necessary to achieve optimal performance.

Thus, the research problem that arises is twofold: Firstly, whether existing CNN models, such as ResNet-18 and AlexNet, can effectively recognize objects from human sketch drawings without significant modifications. Secondly, whether the novel approach of Vision Transformers (ViTs) can outperform or complement CNNs in the domain of sketch recognition, potentially offering insights into the adaptability of self-attention mechanisms to abstract visual representations.

Addressing these questions not only sheds light on the fundamental differences between traditional images and human sketches but also paves the way for the development of specialized models tailored for sketch recognition tasks. Our research aims to provide actionable insights for advancing the field of computer vision and enabling practical applications in areas such as human-computer interaction, digital content creation, and augmented reality.

2 Background Study

The application of sketch analysis includes many uses, such as sketch segmentation, sketch retrieval, and sketch recognition. In our research, we are mainly working with sketch recognition. The main objective is to accurately identify and classify sketches of objects drawn by humans into the proper categories. Sketch recognition has started it's journey in the field of research by Sutherland and Edward (1964). After that, many researchers around the world introduced a variety of methods aimed at improving the accuracy and efficiency of sketch recognition.

Initial studies revealed that humans can recognize sketches with an accuracy of 73.1%, as demonstrated by Eitz et al. (2012). In response, researchers developed computational methods, including a bag-of-features sketch representation and multi-class support vector machines, which achieved a 56% accuracy rate in identifying unknown sketches (Eitz et al., 2012). Various researchers have explored handcrafted features for sketch recognition; notably, a Fisher vector spatial pooling (FV-SP) approach by Schneider and Tuytelaars (2014) has enhanced sketch recognition performance to 68.9% which is close to the human accuracy. These early methods,often lacked the ability to effectively handle the complex and abstract nature of sketches, leading researchers to turn to deep-learning-based approaches.

Deep learning has revolutionized many aspects of computer vision, including sketch recognition. By eliminating the need for manual feature engineering, the introduction of deep learning strategies has transformed how sketches are recognized and analyzed. A study by Yang and Hospedales (2015) presented a deep neural network (sketch-DNN) specifically designed for sketch recognition. This model surpassed traditional approaches by directly learning from raw sketches, achieving a accuracy of 72.2%.

Sarvadevabhatla and Babu (2015) utilized two popular Convolutional Neural Networks (CNNs), ImageNet and a modified version of LeNet, to develop a framework for freehand sketch recognition that leverages ’deep’ features extracted from these CNNs, resulting not in a noticeable improvement in recognition results.

The introduction of Sketch-a-Net by Yu et al. (2015) marked a significant milestone as the first deep learning model specifically designed for freehand sketch recognition to surpass human accuracy. Subsequent modifications to the model, as detailed in (Yu et al., 2016), further enhanced sketch recognition accuracy from 74.9% to 77.95%. Further advancements in the field led to the introduction of the Hybrid CNN by Zhang et al. (2020), which effectively addresses the challenge of recognizing both the appearance and shape features of sketches. This model utilizes a dual-stream architecture consisting of the Appearance CNN (A-Net) and the Shape CNN (S-Net) and achieved a notable accuracy rate of 84.42%. Overall these neural networks methods showed substantial improvements over traditional algorithms.

It is also notable that during 2021 the transition to transformer-based models began, highlighted by the introduction of AttentiveNet (Parihar et al., 2021) and Vision Transformers (ViT) (Wang et al., 2021) for sketch recognition. These models addressed the limitations of CNNs by incorporating elements like dynamic token numbers and attention mechanisms for better classification and efficiency. Recent studies focused on enhancing object localization in sketches using transformer-based models, demonstrating improvements in performance and generalization to new object categories (Tripathi et al., 2023)

3 Research Hypothesis

Our research question is ”How does the accuracy rate of Convolutional Neural Network (CNN) models compared to Vision Transformer (ViT) models vary in classifying objects from human-drawn sketches of objects?”. The CNN models selected for this study include AlexNet, ResNet-18, and

Sketch-A-Net. We selected these models because AlexNet serves as a baseline, offering insight into the performance of earlier CNN architectures and highlighting the evolution of neural network models, Sketch-A-Net is the first tailored CNN model for sketch recognition, which outperforms the human-level accuracy rate Yu et al. (2015), and ResNet-18, a modern CNN architecture with residual connections, provides a deeper network that can capture more complex features, helping us evaluate the impact of network depth on sketch classification.

For the ViT models, we have chosen ViT-B/32, ViT-L/32, and the larger ViT-H/14 to explore different sizes of vision transformers. ViT-B/32 serves as a small-scale model, useful for understanding the baseline performance of ViTs. ViT-L/32, being larger, offers insights into the performance of a more parameter-rich ViT, while ViT-H/14, the largest of our selected models, is expected to capture the most complex details from the sketches due to its fine-grained tokenization.

We hypothesize that the recognition accuracy will vary among the models, with the ViT-H/14 model anticipated to have the highest accuracy, followed by ViT-L/32, ViT-B/32, ResNet-18, Sketch-A-Net, and AlexNet, in descending order. This hypothesis is based on the unique ability of Vision Transformers to capture global context and relationships within images. Unlike convolutional neural network (CNN) architectures, which typically rely on local feature extraction, ViT models tokenize the entire input image and process it as a sequence of tokens. This mechanism allows ViT models to consider the holistic context of the sketch, enabling them to capture the spatial relationships between various components of the object.

We will test the hypothesis by using a dataset consisting of human sketches of objects in order to evaluate and compare the accuracy rates of CNN models, as well as the Vision Transformer models.

4 Research Utilities

In our research, we have used the following utilities and technologies to ensure the efficiency and accuracy of our study:

Python: We utilized Python as our main programming language of choice for this research project because of its effective libraries and suitability to tasks involving data processing and machine learning. Also, in most of our courses, we have worked with Python, so it will be easier for us as well.

Tensorflow/PyTorch: We have used these powerful open-source machine learning frameworks for model development and training due to their flexibility, ease of use, and strong community support. TensorFlow and PyTorch support both CNN and Vision Transformer models.

IA@UGA GPU platform: We used the IA@UGA GPU platform, equipped with NVIDIA RTX 6000 GPUs, to efficiently run our deep learning models. This platform includes both TensorFlow and PyTorch, allowing us to use these powerful frameworks to optimize the execution of complex algorithms and train effectively. This setup, powered by advanced NVIDIA drivers and CUDA technology, ensures high-performance computing for our research.

Latex: We have used Latex for our documentation. It has many advantages, including excellent typesetting, the ability to write mathematical notation, automatic features like numbering, and—above all—effective citation and bibliography formatting through the use of BibTeX.
These utilities have helped us a lot to complete our project.

5 Methodology

5.1 Dataset

The dataset used in our study, referred to as the ”TU-berlin,” consisted of 20,000 sketches gathered from publicly available internet platforms, with a focus on abstract representations. Following the methodology outlined by Eitz et al. (2012) at Technische Universität (TU) Berlin, participants contributed sketches via Amazon Mechanical Turk (AMT), resulting in 22,500 Human Intelligence Tasks (HITs) submitted. Each HIT requested 90 sketches per category across 250 predefined categories, ensuring a diverse and comprehensive representation of objects. The dataset involved 1,350 unique participants investing a total of 741 hours, with a median drawing time per sketch of approximately 86 seconds, reflecting variations in drawing speeds. Rigorous manual inspection and cleaning procedures were then implemented to ensure data quality and consistency, resulting in the exclusion of approximately 6.3 % of sketches due to inaccuracies or deviations. Additionally, each category was refined to contain precisely 80 sketches, aligning with the documentation provided for the TU Berlin Sketch Dataset.

Subset Dataset: 50 categories were meticulously selected based on human recognition capabilities to ensure a diverse representation of everyday objects and concepts. To optimize clarity and precision, five sketches were manually removed from each category, resulting in a final dataset of 3,750 sketches. The decision to remove sketches aimed to eliminate ambiguities and outliers that could disrupt model performance during training and evaluation. This process was essential for improving the overall quality and reliability of the dataset. Additionally, considerations were made regarding GPU limitations, as the computational resources available for model training were finite. The selection of 50 categories and the removal of sketches were influenced by the need to optimize GPU usage and ensure manageable computational loads.

Data Preprocessing

Resizing images to 224x224 pixels aligns with the design of all the models, balances detail and computational load, and leverages empirical evidence of effectiveness. Data augmentation techniques like random horizontal flips and rotations further enhance the model's ability to generalize by simulating real-world variations in sketch data. Converting to tensors and normalizing ensures that the data is in the appropriate format and scale for efficient training in PyTorch.

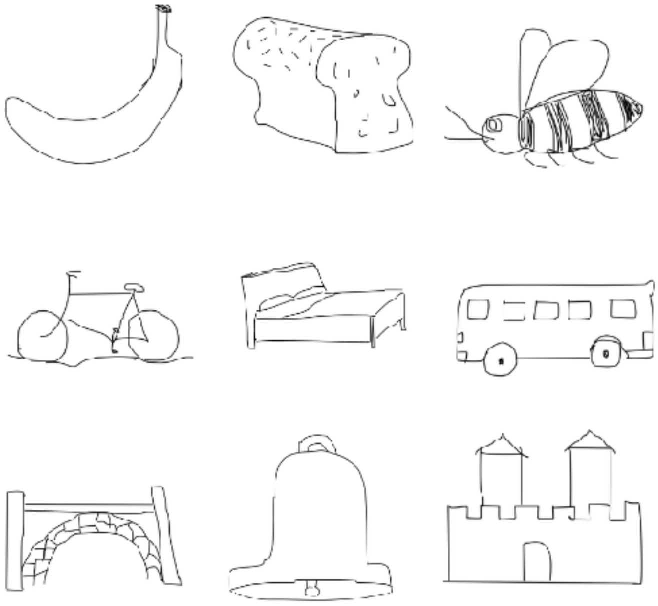


Figure 1: Extract of sketches from TU-Berlin Dataset

5.2 Model Selection

CNN Models: In Convolutional Neural Network (CNN) architectures, the input image undergoes convolutional operations, where small filters are applied to different parts of the image to extract features. These features represent specific characteristics like edges, textures, or complex patterns. The CNN learns to recognize these features by adjusting the weights of the filters during training. Subsequent pooling layers reduce the size of the feature maps while preserving the most relevant information. Finally, fully connected layers combine these features to make predictions about the input image, allowing tasks like image classification, object detection, and image segmentation.

- AlexNet: A deep learning model, introduced by Krizhevsky et al. (2012), known for its effectiveness in image classification tasks. AlexNet contained eight layers; the first five were convolutional layers, some of them followed by max-pooling layers, and the last three were fully connected layers.

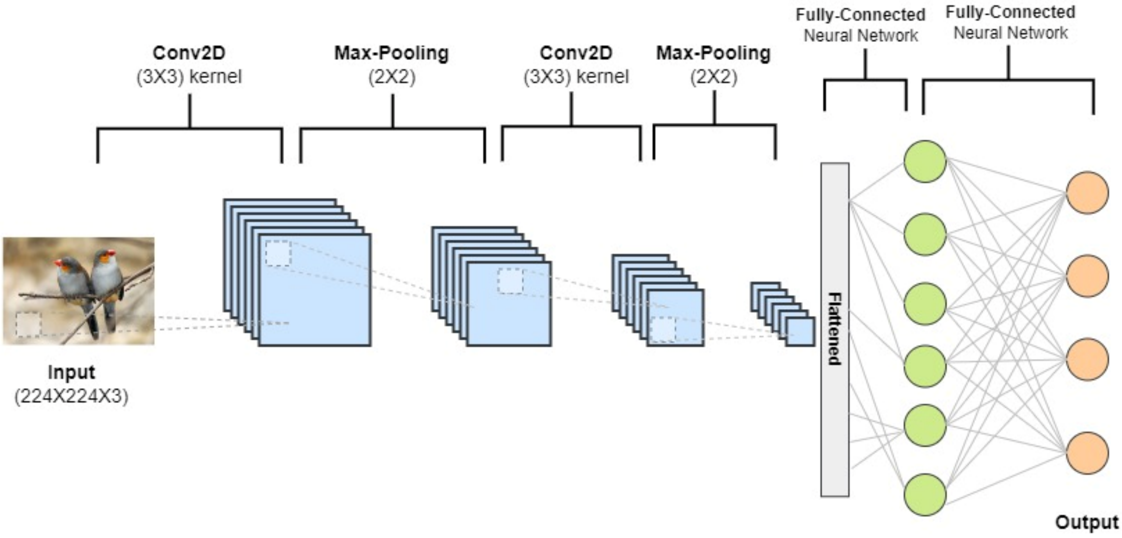


Figure 2: Example of a CNN architecture Maurício et al., 2023

- Sketch-a-Net: Proposed by Yu et al. (2015), this specialized CNN model was made up for sketch recognition tasks.Its architecture comprises five convolutional layers with rectifier (ReLU) units, max pooling, followed by two fully connected layers, and a final layer with 250 output units corresponding to the unique classes in the TU-Berlin sketch dataset, demonstrating a common design pattern in contemporary recognition networks.Yu et al., 2016

- ResNet-18: It is a CNN architecture, introduced by He et al. (2015) in their paper “Deep Residual Learning for Image Recognition.” It renowned for its depth and the use of residual learning blocks.These residual blocks contain skip connections, which allow the network to bypass one or more layers, mitigating the vanishing gradient problem and enabling the training of very deep networks. ResNet-18 excels at extracting hierarchical features from input images.

Vision Transformer: The Vision Transformer (ViT) architecture, introduced by researchers from Google Research, Brain Team, Dosovitskiy et al. (2021), represents fixed-size tokens of specific regions of an input image. These tokens are generated by dividing the image into smaller, non-overlapping regions and linearly projecting each region. These tokens serve as the input to the transformer model, enabling it to process image data in a sequence-like format. Embeddings are vector representations of these tokens, capturing their features and characteristics, and are learned during training to encode meaningful information about the input image. Patches refer to the segmented portions of the input image extracted for processing, with each patch typically containing a portion of the overall visual information. By processing these patches individually, the ViT model can analyze different regions of the image and extract features, facilitating image recognition tasks with varying levels determined by the patch size.

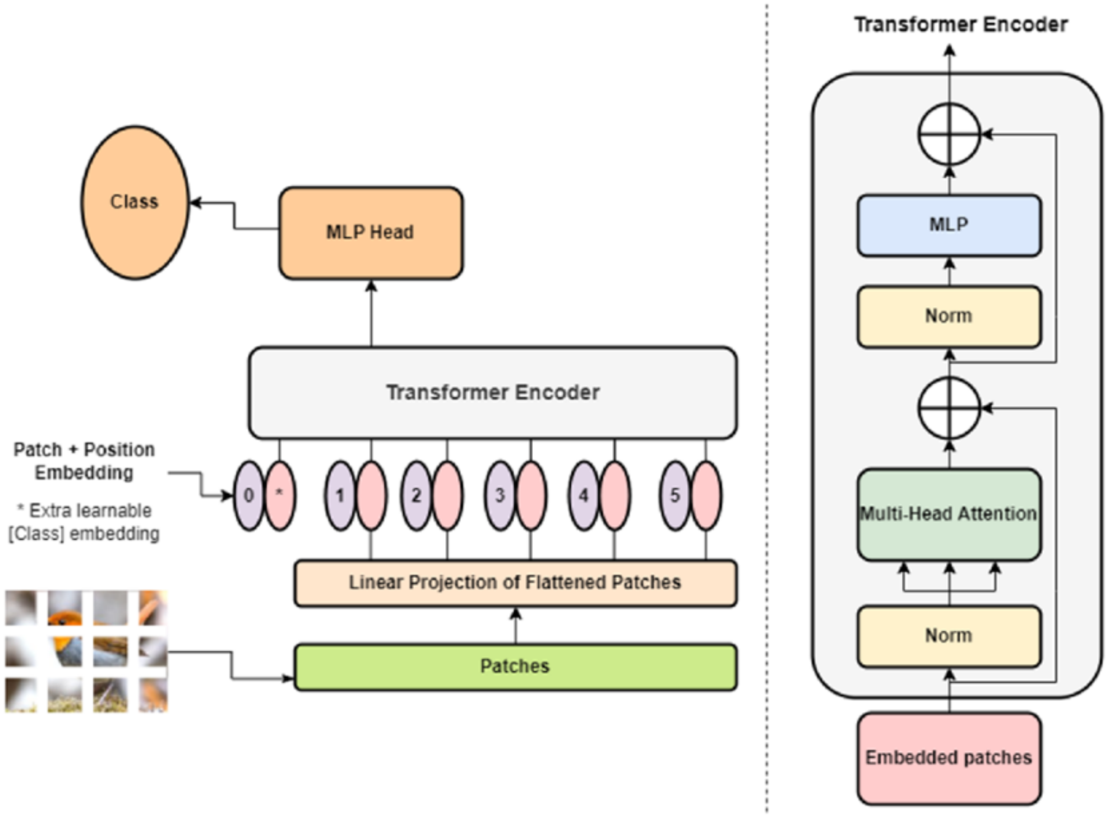


Figure 3: Example of a ViT architecture Maurício et al., 2023

- ViT-B/32: a middle ground between the previous 2 configurations with respect to computational cost and amount of expressive detail captured by ImageNet due to it’s 32x32 patch size.

- ViT-L/32: a large transformer with 32x32 patches, which performs well on fine-grained recognition (like recognizing different species of birds).
- ViT-H/14: excelled at extracting fine-grained details thanks to the 14x14 patch size, but given its depth and capacity it is correspondingly more expensive.

Pretrained Models and Sketch-a-Net

Pretrained models have been previously trained on large datasets such as ImageNet. These models have already learned a broad set of features, which allows them to adapt quickly to new tasks with fewer training iterations. For our experiment, we used the pretrained models were fine-tuned for 10 epochs. Given their pre-existing knowledge from extensive training on diverse datasets, pretrained models only needed a limited number of epochs to adjust to our specific sketch dataset. Fine-tuning these models allows them to transfer their learned features to the new task effectively. Sketch-a-Net, unlike the pretrained models, was trained from scratch. Training from scratch means the model starts with randomly initialized weights and learns all features directly from the sketch dataset. Sketch-a-Net required 50 epochs to achieve satisfactory accuracy. Without pre-existing feature knowledge, Sketch-a-Net needed significantly more epochs to learn the relevant features from the sketches. This extended training period allows the model to iteratively adjust its weights, gradually improving its ability to recognize and classify the sketches.

Train-Test-Validation Split

By using a stratified split, we ensured that the distribution of categories was consistent across the training, validation, and testing sets. This approach was essential for verifying that our models were effectively learning and generalizing across all categories in the dataset. The train-test function was employed to verify that all categories were included in the training process. This function ensures that each category is adequately represented in the training, validation, and testing subsets, thereby maintaining the integrity of the model evaluation. This method helps in preventing any bias that might arise from uneven category distribution, ensuring that the models are trained and evaluated on a comprehensive representation of the data.

5.3 Evaluation Metrics

The accuracy rate, calculated as the proportion of correctly classified sketches to the total number of sketches, served as the primary metric for model performance.Each model was evaluated based on the accuracy rate and loss values obtained after training for 10 epochs.

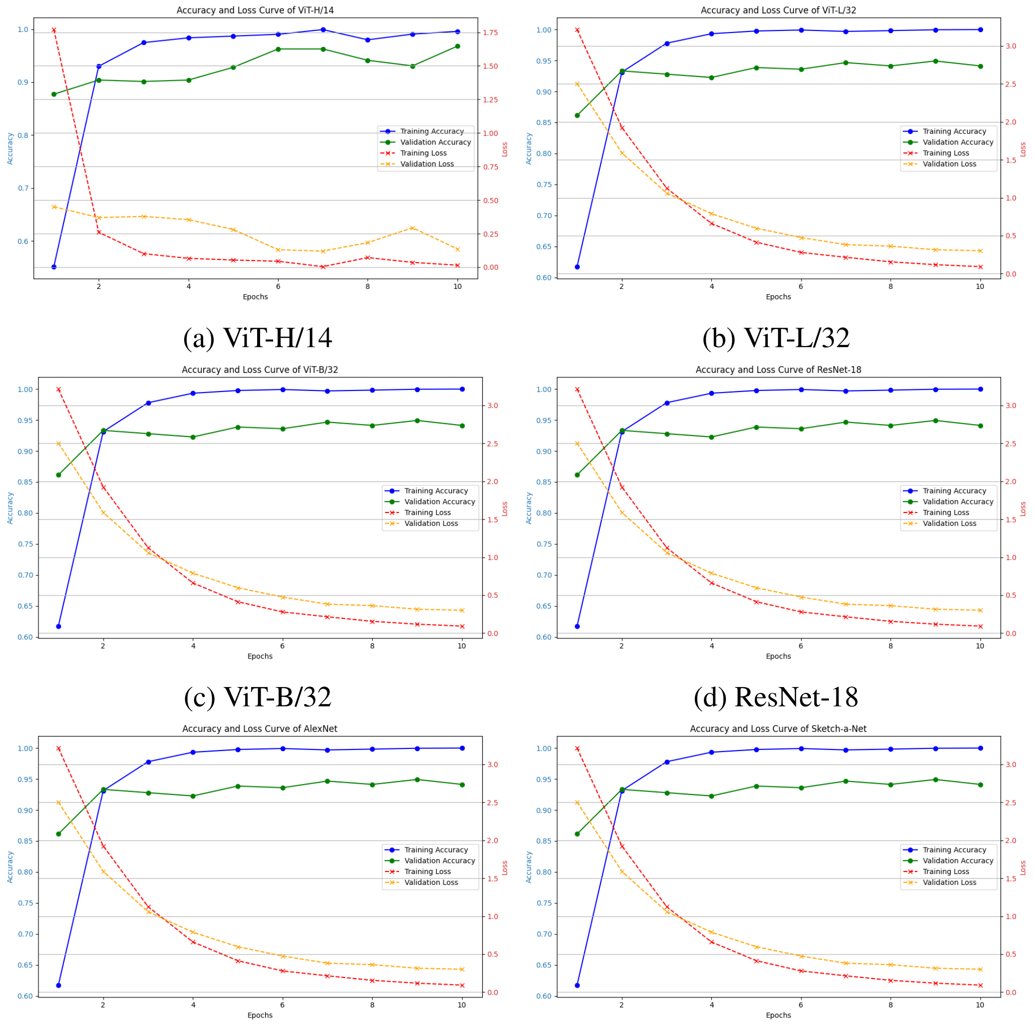
- Top-1 Accuracy is the percentage of predictions where the model's top choice for a given input matches the actual label. In other words, it measures how often the model correctly predicts the most likely class out of all the possible classes.
- Top-5 accuracy is the percentage of predictions where the actual label is among the model's top 5 predictions for a given input. It measures the model's ability to include the correct label in its top 5 predictions, even if it's not the top choice.
- Confusion Matrix is a table that visualizes the performance of a classification model. It shows the number of correct predictions and misclassifications for each class in the dataset. The rows represent the actual classes, while the columns represent the predicted classes.
- Runtime refers to the time it takes for the model to train on the dataset and make predictions. It measures the efficiency and speed of the model's training and inference processes.

6 Results

Comparison of the performance of VIT and CNN models

Accuracy and Loss Analysis

from the below figure 4 ,the ViT-H/14 model performed outstandingly by yielding the highest final test accuracy of 95.73% and the lowest test loss of 0.2024. Therefore, the model possesses the capability to classify test data accurately with minimal error and illustrates that it learned substantially and can generalize properly. Additionally, the ViT-L/32 and ViT-B/32 models closely follow, recording final test accuracy levels of 94.40% and 93.87% and comparability lower test loss. In terms of test accuracy but only slightly underperforms compared to the ViT models, the ResNet-18 model recorded final accuracy of 92.29% and retained comparable test loss. In contrast, the AlexNet model demonstrated average performance with a final accuracy of 89.38% and slightly above test loss, as it exhibited a slightly increased error rate during prediction. Remarkably, the Sketch-a-Net model recorded the lowest test accuracy of 62.33% and the highest test loss, implying that the model would likely struggle with classifying test data accurately. Therefore, the ViT models, especially the ViT-H/14 model, outperformed the rest of the models concerning accuracy and loss metrics.



(a) ViT-H/14

(b) ViT-L/32

(c) ViT-B/32

(d) ResNet-18

(e) AlexNet

(f) Sketch-a-Net

Figure 4: Accuracy and Loss Evaluation of all the selected models

Top accuracy-1 and Top accuracy-5

AlexNet performs well for its first prediction, and even better with the top-5 prediction. Moreover, Sketch-a-Net depicts only the moderate performance of 62.33% top-1 accuracy and 88.53% top-5 accuracy Vision Transformer models (ViT-B/32, ViT-L/32 and ViT-H/14) present the best performance in all settings. The highest top-1 accuracy is achieved by ViT-L/32, 94.40%, and the one for the top-5 is a result of ViT-H/14 with 99.73%. These models show exceptionally high top-5 accuracies, which

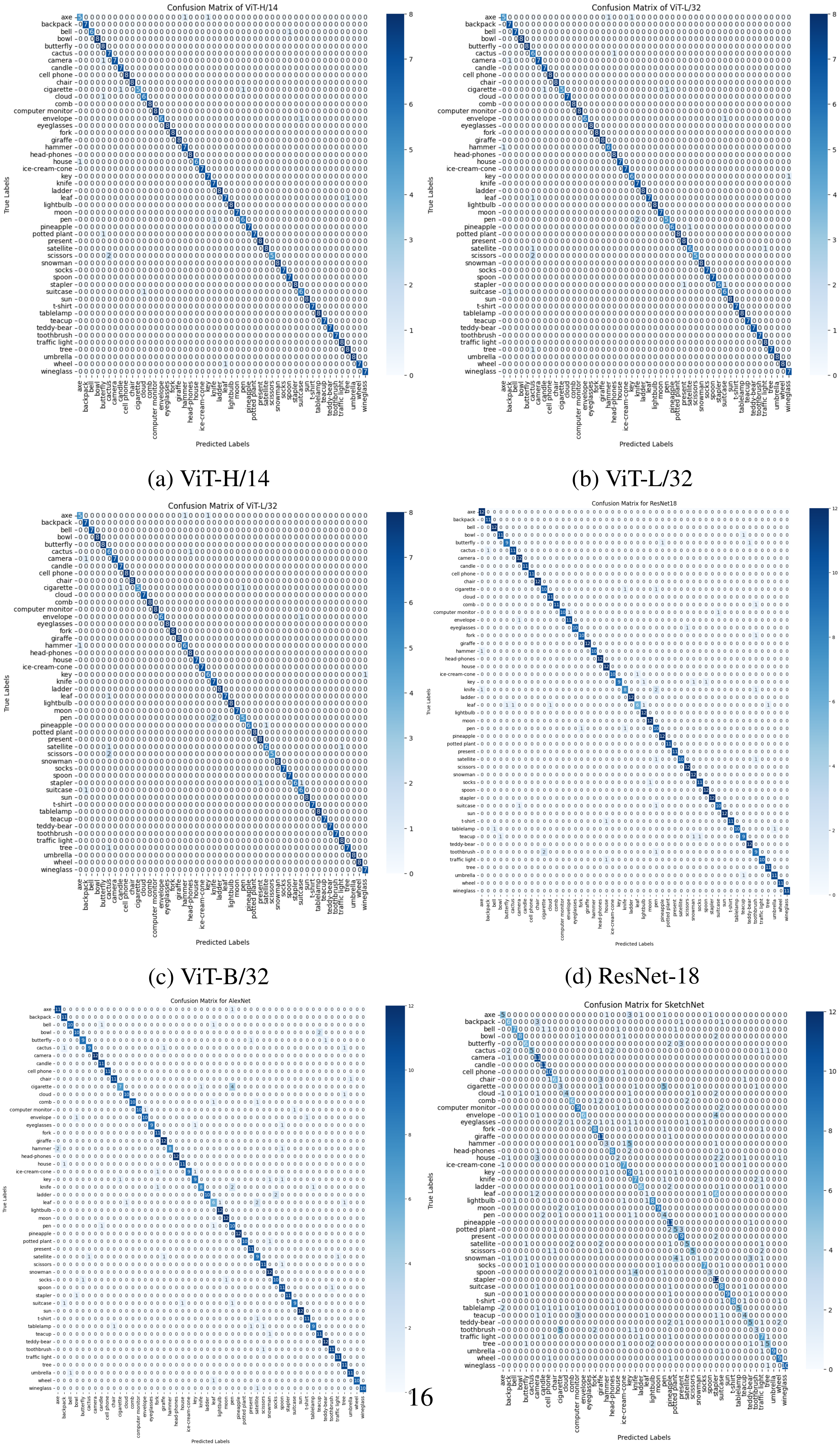
Model	Top accuracy-1 (%)	Top accuracy-5 (%)
Sketch-a-Net	62.33	88.53
Alexnet	89.38	98.12
Resnet-18	92.29	98.97
ViT-B/32	93.87	99.73
ViT-L/32	94.40	99.47
ViT-H/14	95.73	99.20

Table 1: Accuracy comparison of different models for Top accuracy 1 and 5

are between 99.20 to 99.73% It can be seen through the table that the models based on Vision Transformer (ViT) have better performance in accuracy than others. ViT models always show the lowest drop from training to testing accuracy, which suggest that they are more robust (Figure 3). The ViT-H/14 model has the highest top-1 accuracy among all the previously introduced ViT models with a value of 95.73% and also acquires an outstanding top-5 accuracy at 99.20%. ViT-L/32 and ViT-B/32 are more correlated, which might reflect that the core of ViTs can be generally empowered with a larger patch size. ResNet-18 also performs good with top-1 accuracy of 92.29% and top-5 accuracy of 98.97% This is the best performing non-ViT model we report in this experiment. To summarize, AlexNet really separates from the rest. Despite being only second best in terms of top-1 accuracy (89.38 %), it consistently performs very well on all datasets. Sketch-a-net is at the bottom for almost all measures which speaks strongly against generalization capabilities. Although all the ViT models perform better than all non-ViT competitors, there are several different popular non-vision transformer architectures that compete reasonably for second place specifically ResNet-18.

Confusion Matrix

Sketch-a-Net shows reasonable performance, with several correct classifications along the diagonal. For instance, it can be seen in Figure 5 (f) the 6 ”butterfly” has been correctly classified. The value ”5” in the row



(a) ViT-H/14

(b) ViT-L/32

(c) ViT-B/32

(d) ResNet-18

(e) AlexNet

(f) Sketch-a-Net

Figure 5: Confusion matrix of all the selected models

corresponding to ”toothbrush” and the column for ”cigarette” shows that the model incorrectly predicted ”cigarette” when the true label was ”toothbrush” in 5 instances. “These errors indicate that Sketch-a-Net can well take some features, but when it comes to categories which are similar or alike pattern, the misclassification rate ups correspondingly,” the group writes.

The confusion matrix for ResNet-18 shows a much stronger performance and a clearer diagonal that demonstrates more correct prediction are made widespread on the property axis. While there were major mistakes like labeling of ”cigarette” as “pen” overall the number off-diagonal values has reduced. The benefits of ResNet-18’s deeper architecture and increased competence at learning highly nuanced features mean that on the whole, it outperforms when both models are appropriately trained to classify images correctly, though this is not without its difficult classes.

AlexNet Using less off-diagonal values rather than Sketch-a-Net, however, better performance is shown with AlexNet. Although it had its share of mistakes (for example ”cigarette” images that were labeled as ”pen”), the rate of accurate classifications was one order of magnitude better.

The better classification performance of AlexNet with more instances proves its power for feature extraction as well as its representation capability, although the architecture is older now compared to nowadays models. The poor classification rate of SketchNet is even worse compared to both ResNet-18 and AlexNet, indicating the superior performance achieved by these two more complex designs.

ViT-H/14, the confusion matrix demonstrates strong image classification capability. It clearly visible in the figure 5(a) diagonal that there are very few off-diagonal true positive values present, these errors are indeed sparse. Such as ”scissors” has been misclassified wiTh ”cactus”. This is because the model was able to capture features and very well differentiate those features. That is why it yields such high accuracy.

Similar to the ViT-H/14, ViT-L/32 has also demonstrates strong results.

The diagonal has high values across the board, which means it correctly classifies a lot of their images. But we can see that ”scissor” has been misclassified with cactus and similar case with ”pen” and ”knife”. Which is a good sign to test it again and check if the model performance was consistent or were they merely outliers.

ViT-B/32 also exhibits high performance, with the majority of its values clustering around the diagonal. It correctly identifies almost all the images, for example showing exactly 8 images of ”buttefly”,”cell phone” as found. We can see that there are still some misclassifications, for example ”knife” has been predicted as ”pen” but the number of errors here is much lower as compared to the CNN models. Just like other versions, the ViT-B/32 model enables important and distinguished features to be captured, which achieve high levels of accuracy.

Runtime Comparison

From figure 6 below, we note that ResNet-18 and AlexNet top the list of the least total runtimes, and the least total runtimes are followed by ViT-B/32 and Sketch-a-Net, while those of ViT-L/32 are further followed with ViT-H/14. That is, ViT-H/14 seems to consume the highest amount of runtime, which suggests that its training and evaluation process take a much longer time. This plot, therefore, helps paint a clear picture of the total computational resources required for each model. This information makes it possible for one to determine whether the computational resources required for a ViT model can be practical and scale it to be a real-life application of the model.

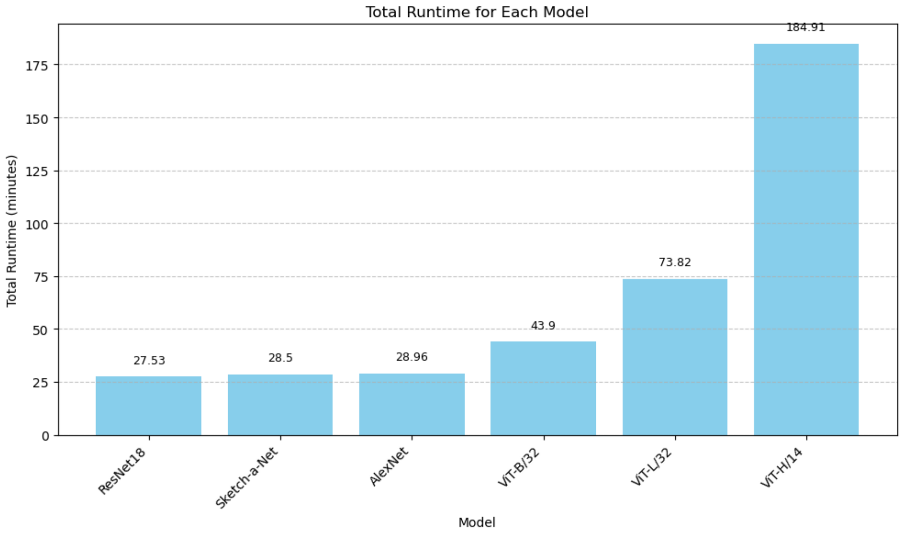


Figure 6: A comparative analysis of runtime durations across all the models

Comparison of the Dataset and the Sub dataset

In this comparison, only Convolutional Neural Network (CNN) models were utilized due to limitations in computational resources. The Vision Transformer (ViT) models were not run on the main dataset containing 250 categories of sketches due to CPU limitations and the need for more GPU resources. Instead, a sub-dataset comprising 50 categories was created to accommodate these constraints. The main dataset consisted of a diverse range of categories, while the sub-dataset aimed to maintain a representative sample while reducing the computational burden.

Model	with (50 categories) Dataset (%)	with (250 categories) Dataset (%)
Sketch-a-Net	62.33	48.10
Alexnet	89.38	70.90
Resnet-18	92.29	74.23

Table 2: Comparison of accuracy of CNN models on the Dataset and the Sub dataset

The results presented in the table 2 above display performance in enhanced accuracy for the 50 categories dataset compared to the model built on the larger dataset of 250 categories. All the three Sketch-a-Net, AlexNet, and

ResNet-18 models exhibited a consistent pattern of a decreased accuracy of between 20-40 % when the dataset was increased to 250 categories. Additionally, the smaller size of the dataset could help alleviate the computational burdens that the models could not perform efficiently due to limitations. Nevertheless, the results obtained demonstrate that with the variations in the datasets, a size test of similar sets of categories would demonstrate considerable stability of selected categories. The findings suggest that an equal number of categories for reduced data are representative of the general data and captured critical features that helped in ensuring outstanding level of performance. The use of a reduced dataset showed excellent potential, even when the computer limitations force the analysts to employ a dataset with 250 categories.

Sketch-a-Net model training
Sketch-a-Net is the only model that needed more epochs to achieve greater accuracy. This can be justified by its architecture and the lack of training compared to the other models that has been mentioned earlier. For 50 epochs, we got 73.63% of accuracy on test set with total run time of 54.30 minutes.This result confirm the importance of the number of epochs.This property refers to the number of times the training dataset is complete covered, A batch size must be at least 1 and no greater than the number of examples in the training dataset.

7 Conclusion and Discussion

In our study, we compared the performance of Convolutional Neural Network (CNN) models (AlexNet, ResNet-18, and Sketch-A-Net) with Vision Transformer (ViT) models (ViT-B/32, ViT-L/32, and ViT-H/14) in classifying objects from human-drawn sketches of objects. Our hypothesis predicted that the ViT-H/14 model would achieve the highest accuracy, followed by ViT-L/32, ViT-B/32, ResNet-18, Sketch-A-Net, and AlexNet. The results of our experiments confirmed our hypothesis regarding the better performance of ViT models. Specifically, the ViT-H/14 model achieved the highest accuracy, followed by ViT-L/32 and ViT-B/32. The experimental results confirmed the superior performance of the ViT models, with the ViT-H/14 model achieving the highest accuracy. This superior performance of ViTs can be attributed to their ability to process the image as a series of patches, enabling them to capture global context within the image more effectively than CNNs. However, contrary to our hypothesis, Sketch-A-Net performed the worst among the models evaluated, with AlexNet outperforming Sketch-A-Net. The observed performance ranking was: ViT-H/14, ViT-L/32, ViT-B/32, ResNet-18, AlexNet, and Sketch-A-Net.

Despite having good results, we acknowledge that there are some limitations to our approach. Due to computational constraints, we limited our dataset to 50 categories from the TU-Berlin dataset. This reduced dataset may not fully capture the diversity and complexity of human-drawn sketches, potentially impacting the generalizability of our findings. Additionally, except for Sketch-A-Net, all models were pretrained. The absence of a pretrained version of Sketch-A-Net may have contributed to its lower performance, as it had to be trained from scratch on our dataset. Furthermore, the limited CUDA memory of our machine constrained the training process, particularly for the ViT models, impacting our ability to utilise the full TU-Berlin dataset effectively.

To overcome these limitations and improve our research, we have outlined the future direction. First, employing the entire TU-Berlin sketch dataset along with other datasets such as QuickDraw and Sketchy to enable a more diverse and extensive evaluation of model performance. Additionally, working with more advanced computational resources with higher CUDA memory would allow for the training of models on larger datasets, facilitating better comparisons. Exploring hybrid models that integrate the strengths of both CNN and ViT architectures could result in superior performance in sketch recognition tasks. Moreover, using a wider range of evaluation metrics, including precision, recall, and F1-score, would provide a more thorough assessment of model performance.

References

Sutherland & Edward, I. (1964). Sketchpad: A man-machine graphical communication system. *Simulation*, *2*(5), R–3.

Eitz, M., Hays, J., & Alexa, M. (2012). How do humans sketch objects? *ACM Trans. Graph. (Proc. SIGGRAPH)*, *31*(4), 44:1–44:10.

Schneider, R., & Tuytelaars, T. (2014). Sketch classification and classification-driven analysis using fisher vectors. *ACM Trans. Graph.*, *33*, 174:1–174:9. https://doi.org/10.1145/2661229.2661231

Yang, Y., & Hospedales, T. M. (2015). Deep neural networks for sketch recognition. *ArXiv*, *abs/1501.07873*. https://api.semanticscholar.org/CorpusID:195345953

Sarvadevabhatla, R. K., & Babu, R. V. (2015). Freehand sketch recognition using deep features. *CoRR*, *abs/1502.00254*. http://arxiv.org/abs/1502.00254

Yu, Q., Yang, Y., Song, Y.-Z., Xiang, T., & Hospedales, T. M. (2015). Sketch-a-net that beats humans. *British Machine Vision Conference*. https://api.semanticscholar.org/CorpusID:15004083

Yu, Q., Yang, Y., Liu, F., Song, Y., Xiang, T., & Hospedales, T. M. (2016). Sketch-a-net: A deep neural network that beats humans. *International Journal of Computer Vision*, *122*(3), 411–425. https://doi.org/10.1007/s11263-016-0932-3

Zhang, X., Huang, Y., Zou, Q., Pei, Y., Zhang, R., & Wang, S. (2020). A hybrid convolutional neural network for sketch recognition. *Pattern Recognition Letters*, *130*, 73–82. https://doi.org/10.1016/j.patrec.2019.01.006

Parihar, A. S., Jain, G., Chopra, S., & Chopra, S. (2021). Sketchformer: Transformer-based approach for sketch recognition using vector images. *Multimedia Tools and Applications*, *80*(6), 9075–9091. https://doi.org/10.1007/s11042-020-09837-y

Wang, Y., Huang, R., Song, S., Huang, Z., & Huang, G. (2021). Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. *ArXiv*. https://doi.org/10.48550/arXiv.2105.15075

Tripathi, A., Mishra, A., & Chakraborty, A. (2023). Query-guided attention in vision transformers for localizing objects using a single sketch.

Maurício, J., Domingues, I., & Bernardino, J. (2023). Comparing vision transformers and convolutional neural networks for image classification: A literature review. *Applied Sciences*, *13*(9). https://doi.org/10.3390/app13095521

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks, 1097–1105.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.