Media Engineering and Technology Faculty
German University in Cairo

Scene-based Searching for Disabled People: Extracting Image Descriptions

Bachelor Thesis

Author: Nihal Samir Mannaa
Supervisor: Dr. Ahmed Abdelfattah
Submission Date: 01 June, 2023

Media Engineering and Technology Faculty
German University in Cairo

Scene-based Searching for Disabled People: Extracting Image Descriptions

Bachelor Thesis

Author: Nihal Samir Mannaa
Supervisor: Dr. Ahmed Abdelfattah
Submission Date: 01 June, 2023

This is to certify that:

(i) The thesis comprises only my original work toward the Bachelor Degree.

(ii) Due acknowledgement has been made in the text to all other material used.

Nihal Samir Mannaa
01 June, 2023

Abstract

In today's digital age, searching for information online has become a ubiquitous part of our daily lives.[10] However, traditional text-based search engines can be challenging for individuals with disabilities or those who do not speak the same language. Furthermore, even for people who can use text-based searches, it can sometimes be difficult to articulate their search query in words.[15] In response to these challenges, scene-based searching has emerged as an alternative approach that allows users to search for information using images instead of text. [13]
This thesis aims to develop a tool that enables people with disabilities and those who do not speak the same language to search for information online through scene-based searching. The tool is designed to extract image descriptions from a scene, providing a more effective means of searching for information compared to traditional text-based searches. The tool, named Scene2Search, will be developed using a set of algorithms and machine learning models that accurately analyze images and extract relevant information.

Contents

Acknowledgments

I would like to express my gratitude to my supervisor Dr. Ahmed Abdelfattah for his invaluable guidance, encouragement, and support throughout my research. His expertise, insightful comments, and constructive feedback have been instrumental in shaping the direction of this thesis.

I would like to thank my family and friends for their unwavering support and encouragement throughout this journey. Their love and understanding have been a constant source of motivation and inspiration.

Finally, I would like to acknowledge the contributions of the various research papers, datasets, and tools used in this study. Without the availability of these resources, this research would not have been possible.

Thank you all for your support and encouragement.

Chapter 1

Introduction

1.1 Motivation

Despite the potential benefits of scene-based searching, existing tools often suffer from limitations such as low accuracy.[10] For example, many image search engines rely on metadata or tags associated with images, which can be incomplete, inaccurate, or not reflect the user's intent.[19]

Therefore, there is a need for a more effective tool that leverages scene-based searching to enable people with disabilities and those who do not speak the same language to search for information online only using sketching.[15]

1.2 Purpose

The purpose of this project is to develop a tool that addresses the limitations of existing scene-based searching tools and provides a more inclusive and accessible alternative for people with disabilities and those who do not speak the same language. The tool, called Sketch2Search, allows users to draw a hand-drawn sketch of the information they are looking for, and then extracts relevant information from the sketch to initiate an online search. By using image recognition and machine learning algorithms, Sketch2Search can accurately analyze hand-drawn sketches and extract relevant information with a high level of precision and recall.

This project aims to develop a tool that can overcome language barriers by enabling users to search for information using images; therefore, Sketch2Search can provide a more inclusive and accessible alternative to traditional text-based searches. The following sections of this thesis will provide a detailed description of the development and evaluation of Sketch2Search, including its functionality, performance, and results.

Chapter 2

Literature Review

Scene-based searching is an emerging field of research that aims to provide an intuitive and accessible way for users to search for information online.[4] Traditional text-based searches can be challenging for people with disabilities or those who do not speak the same language. Scene-based searching overcomes these limitations by enabling users to search for information using images or sketches. Sketch2Search is a novel tool that leverages machine learning algorithms to enable users to search for information by drawing a hand-drawn sketch of the information they are looking for. In this literature review, we will discuss the state of the art in scene-based searching and machine learning algorithms that underpin the functionality of Sketch2Search.

2.1 CNNs

2.1.1 What is machine learning?

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.[11]

The process of machine learning involves the following steps:

- Data collection: Gathering relevant data that is representative of the problem you want the machine learning model to solve.
- Data preprocessing: Cleaning and transforming the data to make it suitable for analysis. This may involve tasks like removing irrelevant or noisy data, handling missing values, and normalizing the data.
- Feature extraction: Selecting or creating meaningful features from the data that capture relevant patterns or characteristics. These features act as input variables for the machine learning algorithm.

- Model selection: Choosing an appropriate machine learning algorithm or model that suits the problem at hand. The selection depends on factors like the type of data, the problem's nature (classification, regression, clustering, etc.), and the available resources.
- Model training: Using the prepared data, the selected model is trained by feeding it the input features and the corresponding desired outputs (in supervised learning) or by allowing it to explore and learn patterns in the data (in unsupervised learning).
- Model evaluation: Assessing the performance of the trained model using evaluation metrics that measure its accuracy, precision, recall, or other relevant measures. This step helps to identify potential issues, such as overfitting (when the model performs well on training data but poorly on new data).
- Model optimization: Fine-tuning the model by adjusting its parameters or exploring different variations to improve its performance on unseen data. This process may involve techniques like hyperparameter tuning, regularization, or ensemble methods.
- Model deployment: Once the model meets the desired performance criteria, it can be deployed to make predictions or decisions on new, unseen data.[17]

2.1.2 Deep Learning

Deep learning is a subset of machine learning that focuses on using artificial neural networks (ANNs) with multiple layers to learn hierarchical representations of data.

- It is inspired by the structure and function of the human brain.
- Deep learning models, known as deep neural networks, can automatically learn and extract complex features from raw data without explicit feature engineering.[20]
- One of the most popular deep neural networks is Convolutional Neural Networks (also known as CNN or ConvNet) in deep learning, especially when it comes to Computer Vision applications.
  - Convolutional Neural Networks (CNNs) are a specific type of deep learning architecture (Shown in **Figure 2.1**) commonly used for processing and analyzing visual data, such as images or videos.
  - CNNs are designed to automatically learn spatial hierarchies of patterns or features by using convolutional layers, pooling layers, and fully connected layers.
  - They are particularly effective for tasks like image classification, object detection, and image segmentation.

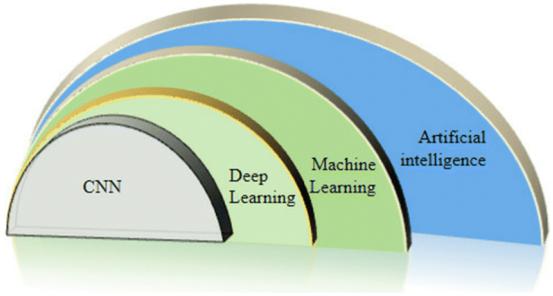

Figure 2.1: The relationship between artificial intelligence, machine learning, deep learning and convolutional neural network[13]

- CNNs leverage the concept of convolution, where filters or kernels slide over the input image, extracting local patterns at different scales. Pooling layers downsample the output of convolutional layers, reducing spatial dimensions while retaining important information.
- Finally, fully connected layers use the extracted features to classify or make predictions about the input data.[13]

The use of CNNs has become increasingly prevalent in recent years[1], thanks to advancements in machine learning algorithms and the availability of large datasets for training these models.

- CNNs have been used for a wide range of applications, from self-driving cars to medical image analysis, and have demonstrated impressive accuracy and performance in many cases.[18]
- In the context of Sketch2Search, the use of machine learning and CNNs enables the tool to accurately recognize and extract relevant information from hand-drawn sketches, providing a more intuitive and accessible search experience for users.
- By leveraging the power of machine learning, Sketch2Search has the potential to overcome language barriers, improve search accuracy, and enhance the overall usability of online information resources.

2.2 Scene-based searching

Scene-based searching has emerged as a promising alternative to traditional text-based searches.

- The basic idea is to enable users to search for information using images instead of text.

- This approach has the potential to overcome many of the limitations of text-based searches, such as language barriers, difficulties in articulating search queries, and limitations in search algorithms.

- Scene-based searching has been used for a wide range of applications, including image retrieval, object recognition, and medical image analysis.[4]

- One of the key challenges in scene-based searching is how to recognize and extract information from images.

- Traditional image retrieval systems rely on metadata or tags associated with images, which can be incomplete, inaccurate, or not reflect the user's intent.

- In addition, these systems often require users to upload images or select pre-existing ones, which can be time-consuming and inconvenient. To overcome these limitations, researchers have turned to machine learning algorithms, specifically deep learning models, which can learn to recognize patterns and features in images and extract information from them.

  - For example, Li et al. [14] proposed a scene-based searching system that uses a CNN to extract features from images and then matches these features with a database of reference images to retrieve similar images. They achieved promising results with an average precision score of 0.7645.
  - Chen et al. [5] developed a scene-based searching system that uses a CNN to recognize objects in images and then generates natural language descriptions of these objects to facilitate search queries. They reported an accuracy of 70.8% in object recognition and an average precision score of 0.7039.

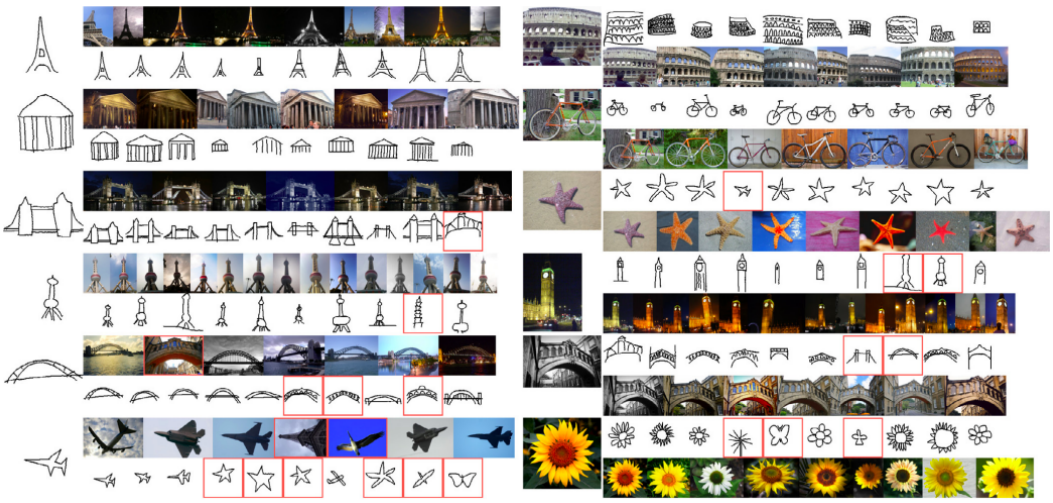

Figure 2.2: Sketch-based image retrieval using CNNs[4]

2.3 Sketch2Search

Sketch2Search is a novel tool that enables users to search for information online by drawing a hand-drawn sketch of the information they are looking for.

- Sketch2Search leverages machine learning algorithms, specifically CNNs, to recognize and extract information from hand-drawn sketches.
- The system works by first converting the hand-drawn sketch into a digital image and then using a CNN to analyze the image and extract relevant features.
- These features are then matched with a database of reference images to retrieve similar images, which are presented to the user as search results.

Sketch2Search has several advantages over traditional scene-based searching systems.

1. It provides a more intuitive and accessible search experience by enabling users to search for information using hand-drawn sketches.
2. It overcomes language barriers and difficulties in articulating search queries by enabling users to search for information using visual representations of their search query.
3. It enhances the overall usability of online information resources by providing a more inclusive and accessible alternative to traditional text-based searches.

Chapter 3

Methodology and Results

3.1 Plan

This is the Gantt I used for my work.

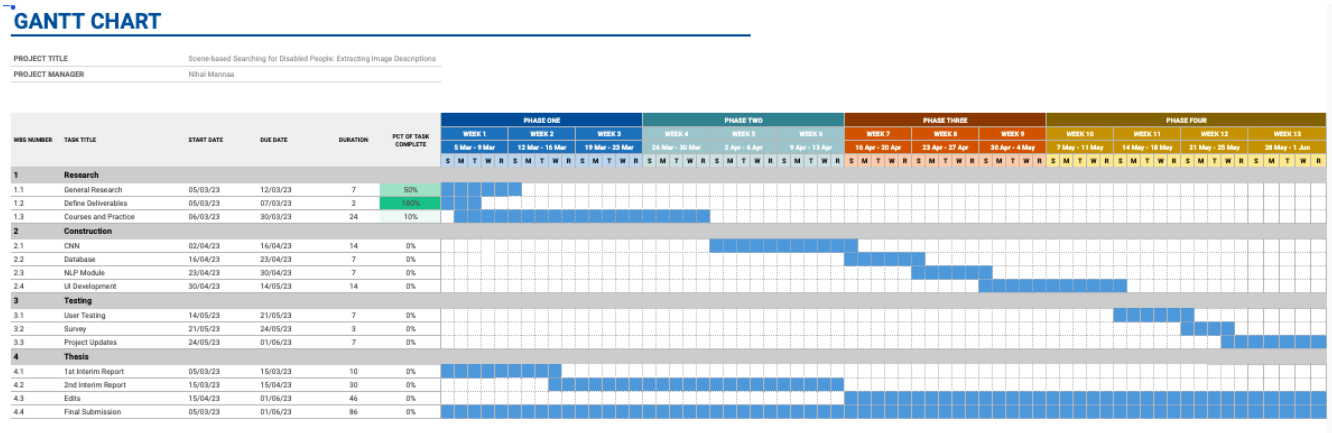

Figure 3.1: Gantt Chart: Timeline

3.2 Quick, Draw! Dataset

The Google Quick Draw dataset[8] was used in this project.

According to its description, the Quick Draw Dataset is a collection of 50 million drawings across 345 categories, contributed by players of the game "Quick, Draw!".[7] The drawings were captured as timestamped vectors, tagged with metadata including what the player was asked to draw and in which country the player was located.

3.3 Technologies Used

Python programming language was mainly used in this thesis, accompanied by the subsequent frameworks/libraries.

3.3.1 Google Colab Pro

Google Colab Pro is a premium version of Google Colaboratory, commonly known as Google Colab. It is a cloud-based platform provided by Google that allows users to run and execute Python code in a browser-based environment. Colab Pro offers additional features that include faster execution times, more powerful hardware resources (such as GPUs or TPUs), longer session durations, priority access to resources, and increased storage space. In this project, GPUs were used in order to optimize the execution time.

3.3.2 TensorFlow

TensorFlow is an open-source machine learning framework developed by Google. It provides a flexible and efficient ecosystem for building and training machine learning models, including deep neural networks. In this project, TensorFlow is used as the underlying backend for executing computations and optimizing the neural network model.

3.3.3 Keras

Keras is a high-level deep learning library that runs on top of TensorFlow. It offers a user-friendly and intuitive interface for building and training neural networks. In this project, Keras is used to define the model architecture, specify the layers, and train the model.

3.3.4 NumPy

NumPy is a fundamental Python library for numerical computing. It provides a powerful array object and functions for manipulating and operating on arrays efficiently. NumPy arrays are multi-dimensional, homogeneous data structures that allow efficient storage and processing of large datasets. In this project, NumPy is used for various operations on the input data, such as loading, reshaping, and normalizing the images.

3.3.5 Matplotlib

Matplotlib is a popular plotting library in Python that provides a wide range of functions and tools for creating visualizations. In this project, Matplotlib is used to plot the training and validation accuracy as well as the training and validation loss of the model. It is also used to display a randomly selected test image.

3.3.6 Google API Client Library

The Google API Client Library is a collection of client libraries that provide easy and convenient access to various Google APIs. It is available for multiple programming languages, including Python, and simplifies the process of interacting with Google services and retrieving data. In this project, the Google API Client Library for Python provides the googleapiclient.discovery module, which enables the project to make requests and retrieve image search results from Google.

3.3.7 IPython library

The IPython library is an interactive computing and development environment for Python. In the project, the 'IPython.display.Image' class is used to display images retrieved from the Google image search results.

3.4 Implementation

3.4.1 Overview

The final version of the code[16] trains a Convolutional Neural Network (CNN) model on the Google Quick Draw dataset to predict hand-drawn sketches. Then, using the Google Custom Search API, a search is performed to find images related to the top 5 predicted sketches.

Out of the 345 categories in the dataset, 45 object categories were chosen for training. These categories include apple, banana, crab, crown, dog, duck, eye, feather, pizza, tree, and more. The model is trained on 2000 samples per object category over 15 epochs.

Shown in **Figure 3.2** is a general idea of the steps taken to complete the project.

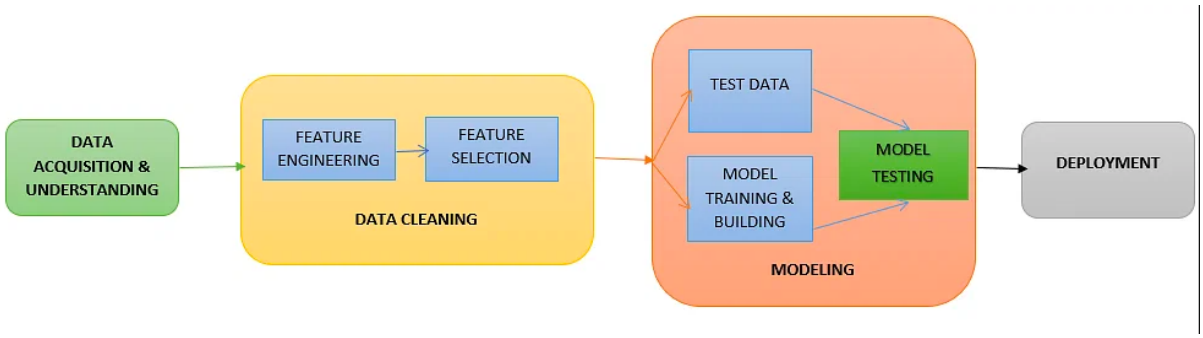

Figure 3.2: Main Steps in the Machine Learning Process[17]

3.4.2 Preprocessing

Data preprocessing in Convolutional Neural Networks (CNNs) is a series of steps taken to prepare the input data before feeding it into the network. These steps aim to enhance the quality of the data, make it compatible with the network's requirements, and improve the overall training and performance of the CNN. This is a description of the preprocessing steps done in this project.

1. **Loading the data:** The sketches for different objects are loaded from the data files using the 'load' function. The sketches are stored as arrays representing the pixel values.

2. **Reshaping and Grayscale Conversion:** The sketches are reshaped to a standard size of 28x28 pixels. Additionally, the sketches are converted to grayscale by expanding their dimensions to (28, 28, 1), where the third dimension represents the grayscale intensity.

3. **Normalization:** The 'normalize' function is applied to the data, scaling the pixel values from the original range of 0-255 to a normalized range of -1 to 1. This step is performed to ensure that all pixel values have a consistent scale and to ensure better convergence during training.

4. **Limiting the number of samples:** The 'set_limit' function is used to limit the number of samples for each object category to a specified value ('N'). This is done to balance the dataset and ensure an equal number of samples for each object during training.

5. **Creating Labels:** The 'make_labels' function generates labels for the samples. Each label is assigned to a specific object category and is repeated 'N' times to match the limited number of samples per category. These labels represent the ground truth for the corresponding sketches.

6. **Train-Test Split:** The preprocessed sketches and their corresponding labels are split into training and testing sets using the 'train_test_split' function. The split is done with a test size of 0.2, resulting in an 80-20 split between the training and testing data, respectively.

7. **One-Hot Encoding:** The labels are converted into one-hot encoded vectors using the 'np_utils.to_categorical' function. This encoding scheme represents each label as a binary vector, where the index corresponding to the object category has a value of 1, and all other indices are 0. This encoding is helpful in ensuring compatibility with numerical computations, avoiding biases, and facilitating multi-class classification tasks.[6]

8. **Data Augmentation:** Data augmentation is performed on the training data using the 'train_datagen' object from 'ImageDataGenerator'. Various transformations,

such as rotation, shifting, shearing, zooming, horizontal flipping, and filling, are applied to artificially increase the diversity of the training data. This helps the model generalize better and handle variations in hand-drawn sketches.

Overall, these preprocessing techniques ensure that the hand-drawn sketch dataset is properly formatted, standardized, normalized, balanced, and augmented to improve the training and generalization capabilities of the CNN model.

3.4.3 Training

1. The CNN model architecture consists of several layers.
    - The first layer is a convolutional layer with 32 filters (feature detectors) and a kernel size of 3x3. It uses the ReLU activation function due to its computational efficiency, non-linearity, and ability to mitigate gradient vanishing. [12]
    - A second convolutional layer follows, with 64 filters and a 3x3 kernel size, activated by ReLU.
    - A max-pooling layer with a pool size of 2x2 reduces the spatial dimensions of the features.
    - A dropout layer is added to prevent overfitting[2], randomly disabling 25% of the previous layer's outputs during training.
    - The output of the dropout layer is flattened to prepare for the subsequent dense layers.
    - A dense layer follows, with 128 neurons, and applies the ReLU activation function.
    - Another dropout layer is included to further regularize the network, randomly deactivating 50% of its outputs during training.
    - The final dense layer employs the softmax activation function to produce predicted probabilities for each object category.
2. Model Compilation and Training.
    - The model is compiled using the categorical cross-entropy loss function, which is suitable for multi-class classification tasks.
    - The Adam optimizer is utilized to optimize the model's weights during training. Accuracy is chosen as the evaluation metric to measure the model's performance.
    - The training process commences, with the model being fit to the training data.
    - The batch size is set to 32, meaning the model updates its weights after processing 32 samples at a time.

- The model undergoes training for 15 epochs, representing the number of complete passes through the entire training dataset.
- The accuracy and the validation accuracy reached 84.09% and 82.81%, respectively.
- As shown in **Figure: 3.3**, both the accuracy and validation accuracy were increasing exponentially, while the loss and validation loss were decreasing exponentially, meaning that the model is not overfitting.

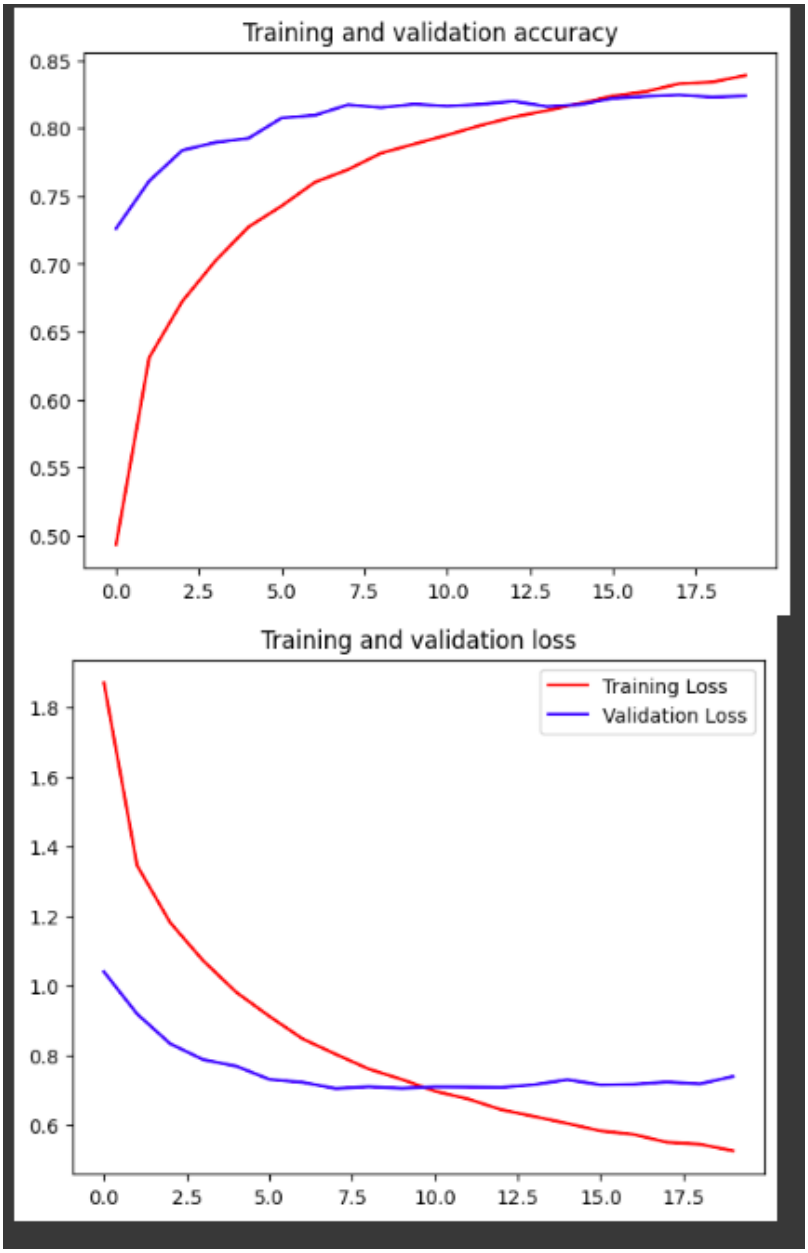

Figure 3.3: Convergence during Training

3.4.4 Testing

1. **Model Evaluation:**

   - After training, the model's accuracy is evaluated using the testing data: the testing accuracy reached 83%.
   - The model has a part that predicts object categories for the testing samples. The accuracy is calculated by comparing the predicted labels to the ground truth labels and calculating the proportion of correct predictions.

2. **Saving the Model:**

   - Once training and evaluation are complete, the trained model is saved for future use. The model is saved in a file format, in this case, with the name "object_recog.h5".

3. **Adding my own Input:**

   - The model has a part for testing that inputs data from one's computer, and tests the model with it.
   - This is the part that struggled the most during the testing phase.
   - The model had to be edited multiple times (changing the number of filters, adding dropout layers, and adding one hot encoding) to get the final satisfactory results.

3.5 Results

The Sketch2Search yielded promising results, demonstrating the effectiveness of the Convolutional Neural Network (CNN) model trained on the Google Quick Draw dataset for predicting hand-drawn sketches. With a testing accuracy of 83%, the model successfully classified sketches of various objects such as apples, bananas, crabs, and more, showcasing its ability to learn and recognize different visual patterns. From **Figure 3.4 to Figure 3.31**, example input sketches and outputs can be shown.

Furthermore, the integration of image search using the Google Custom Search API (shown for the example syringe drawing in **Figure 3.32**) provided 4 visual representations (shown from **Figure 3.34** to **Figure 3.38**) of each of the top 5 predicted sketches (shown in **Figure 3.33**), enhancing the overall user experience.

These outcomes highlight the potential of CNNs in accurately interpreting hand-drawn sketches and generating relevant visual outputs.

Figure 3.4: Pizza slice drawing

3.6 Limitations

- The accuracy of the model heavily relies on the quality and diversity of the training data. If the dataset used for training is limited in terms of variations in sketch styles, angles, or quality, the model's performance may be affected. Consequently, the model's ability to generalize to unseen sketches or handle variations in drawing styles outside the dataset may be limited.

- The restricted utilization of a large number of categories due to limited computational resources, such as memory and processing power, can be another limitation. This constraint hinders the model's capability to effectively recognize and classify a wide range of objects, potentially impacting its overall performance and coverage.

- Moreover, the reliance on image search results, which can be influenced by the search algorithm and may not always provide the most accurate or relevant images for each predicted sketch, could be another limitation.

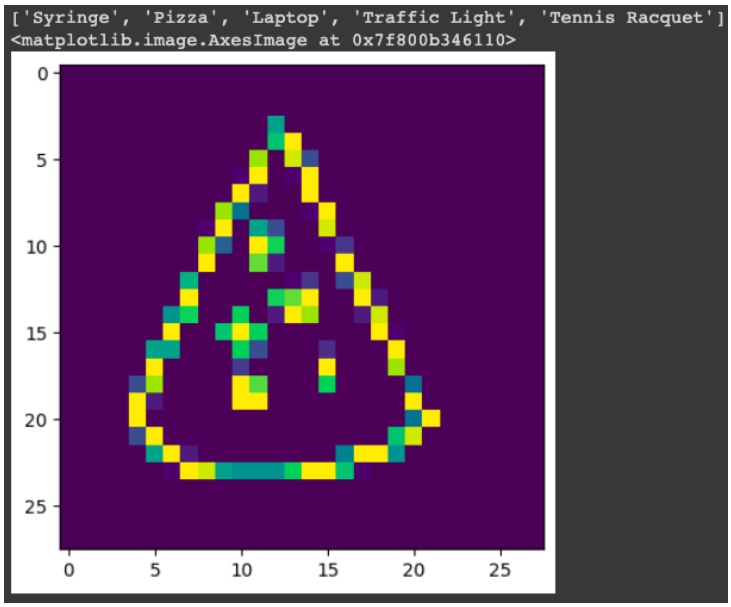

Figure 3.5: Pizza slice predictions

3.7 Comparison to Other Projects

3.7.1 Building a Pictionary App[9]

- A project focused on implementing and training deep learning models for the Quick, Draw! dataset using the PyTorch framework.
- It has a sketchpad as input, and an output list of the 5 top predictions.
- Unlike the Sketch2Search project, it doesn't have output images, so it might not be user friendly for some people.
- It uses 100 classes, while Sketch2Search uses 45 classes.
- Moreover, it has a lot of random guesses from the sketch input. (**Figures 3.39 to 3.41**).

3.7.2 Quick Draw by Akshay Bahadur[3]

- Only 15 categories used, so Sketch2Search has more variety in the object recognition.
- Output emojies rather than multiple Google search images.

Figure 3.6: Tree

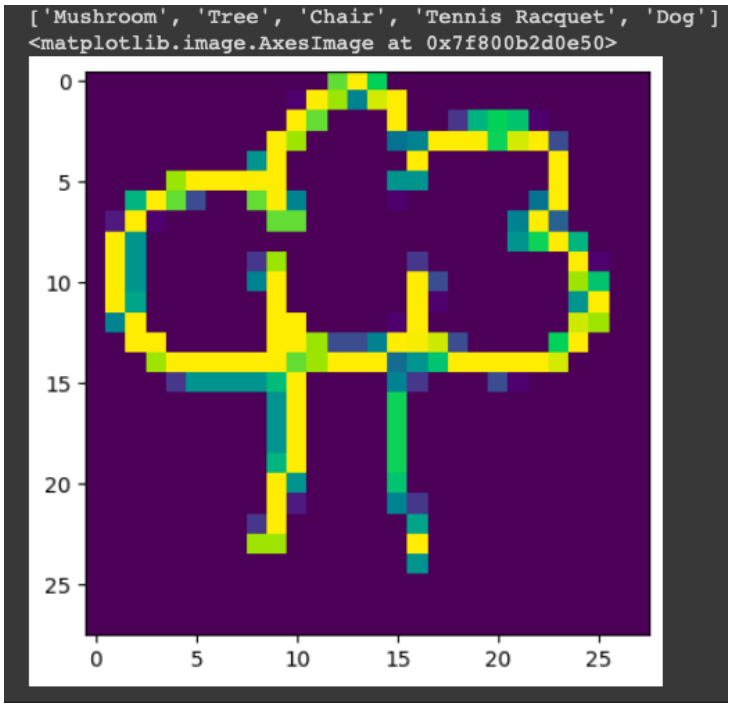

Figure 3.7: Tree predictions

Figure 3.8: Duck

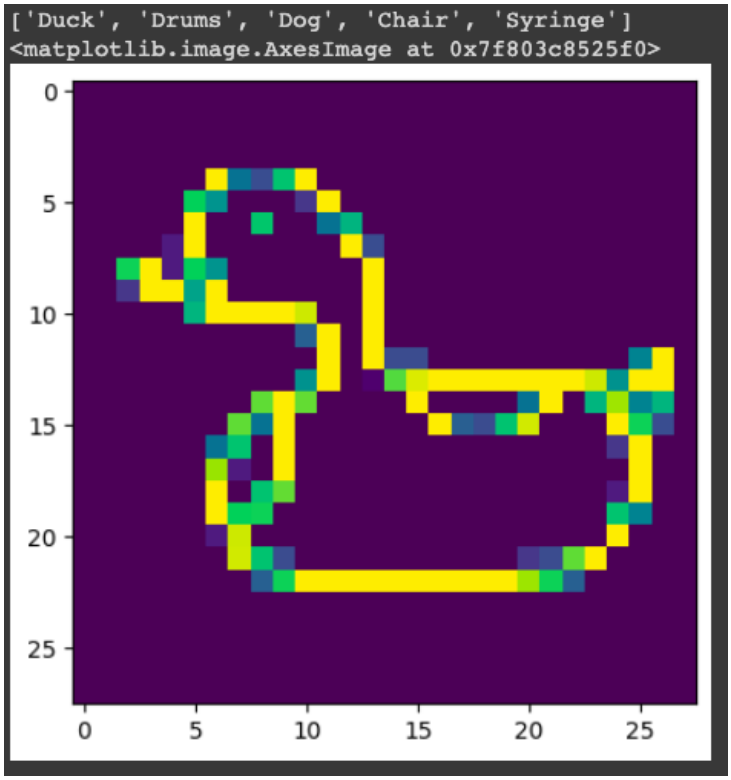

Figure 3.9: Duck predictions

Figure 3.10: Feather

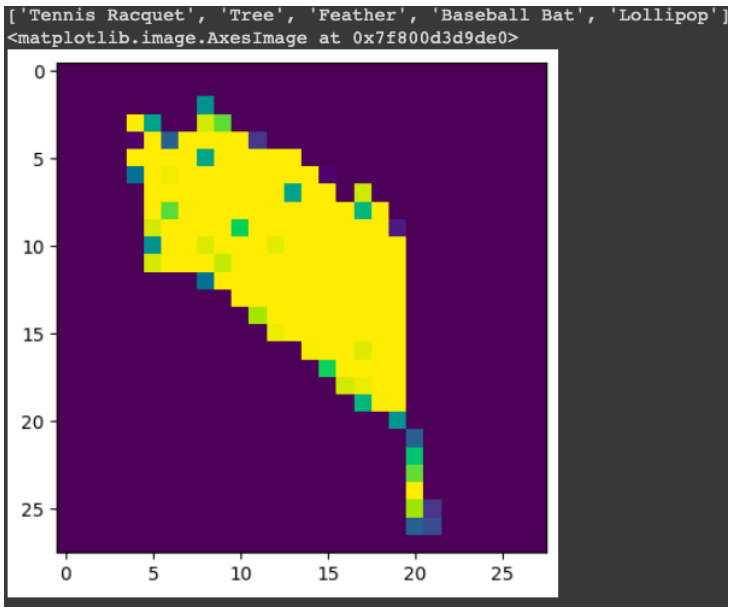

Figure 3.11: Feather predictions

Figure 3.12: Umbrella

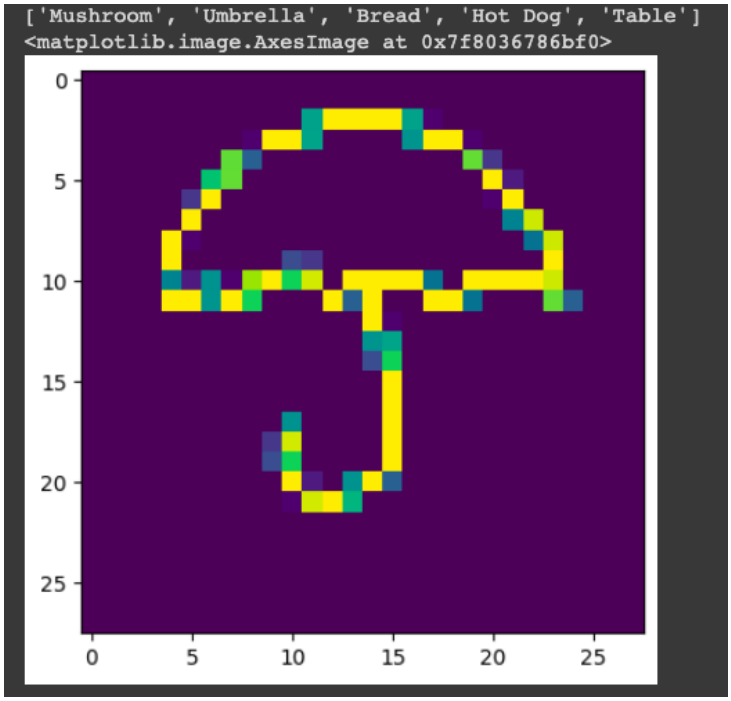

Figure 3.13: Umbrella predictions

Figure 3.14: Feather

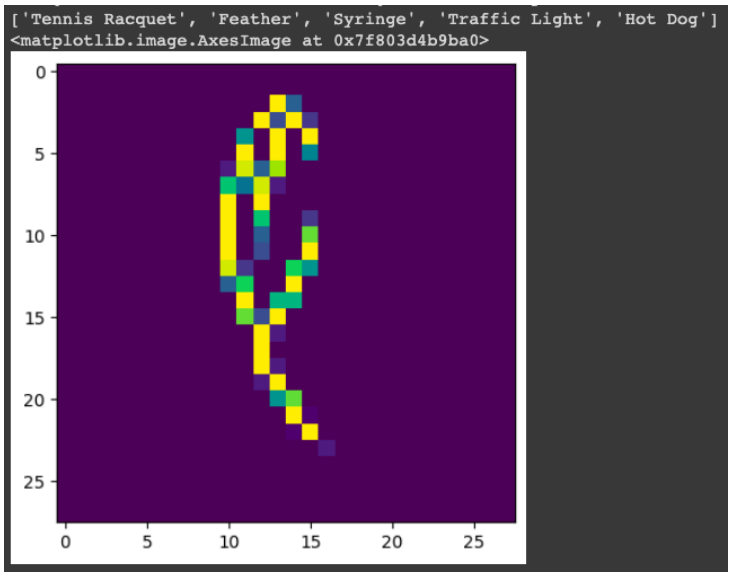

Figure 3.15: Feather predictions

Figure 3.16: Book

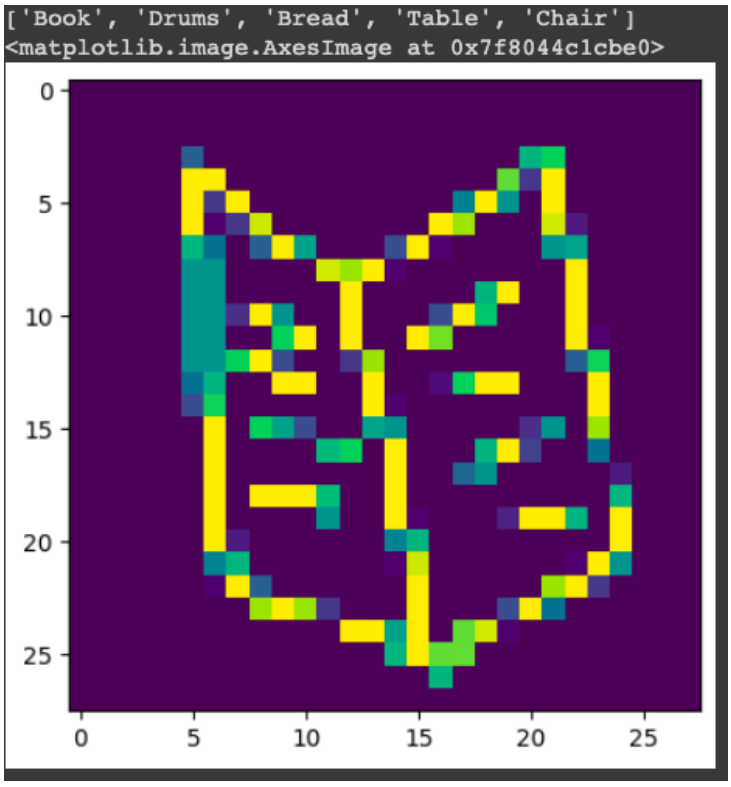

Figure 3.17: Book predictions

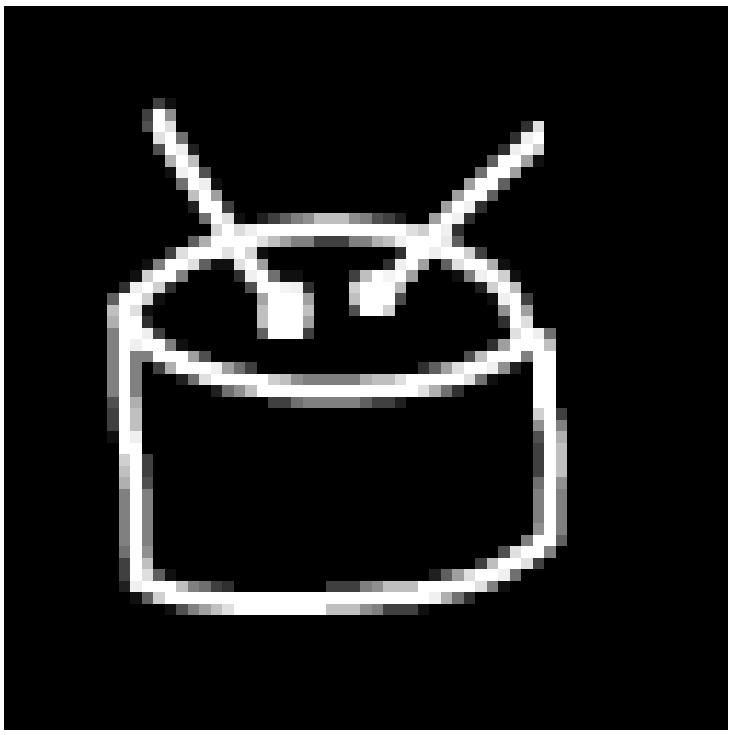

Figure 3.18: Drums

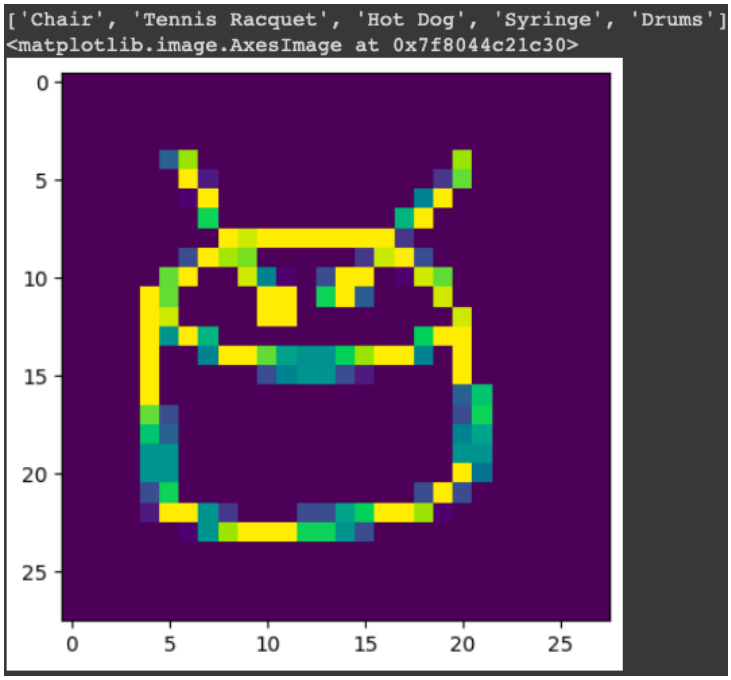

Figure 3.19: Drums predictions

Figure 3.20: Dumbbell

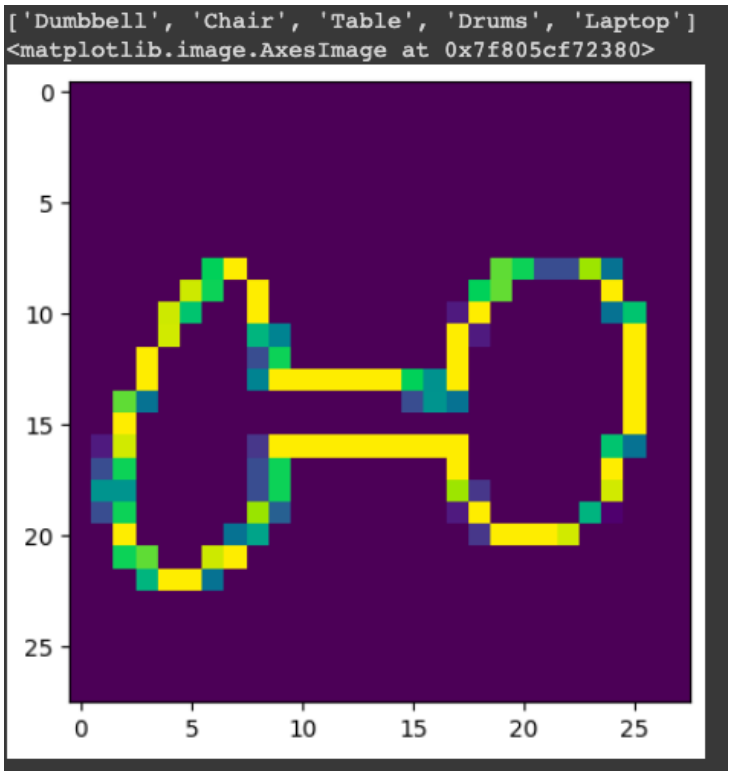

Figure 3.21: Dumbbell predictions

Figure 3.22: Pants

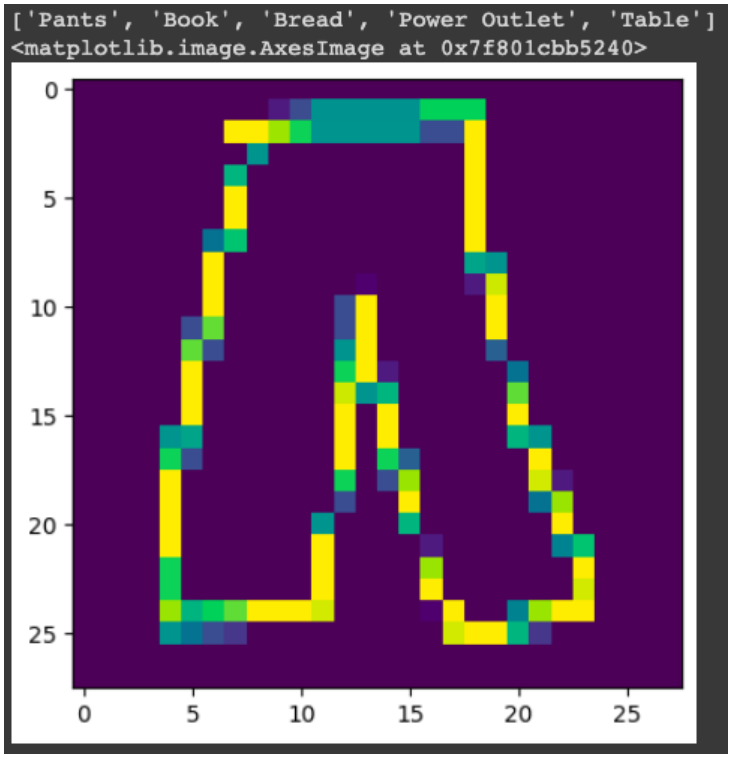

Figure 3.23: Pants predictions

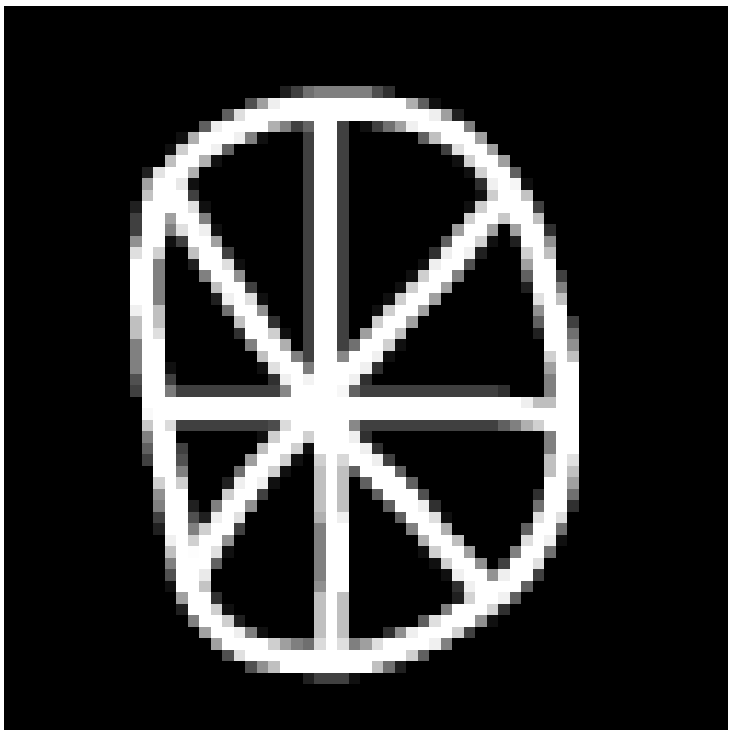

Figure 3.24: Pizza

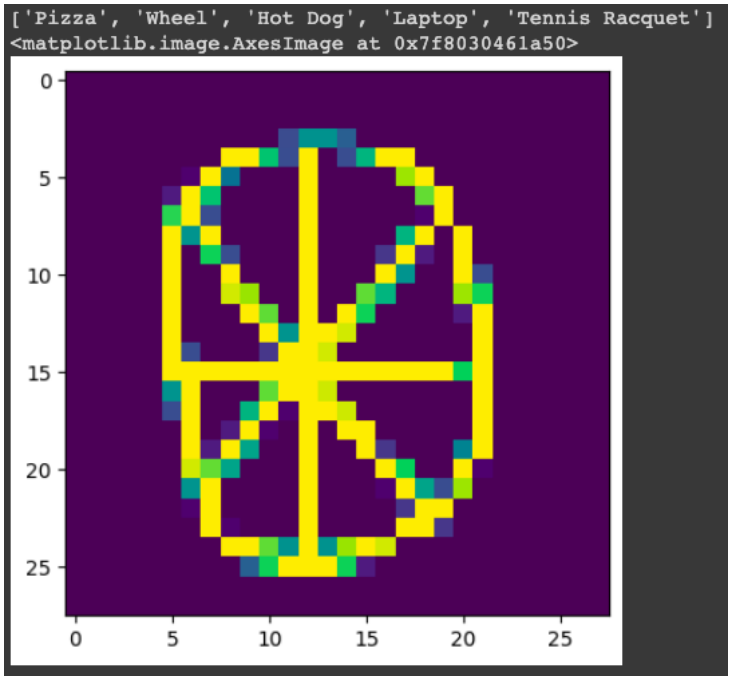

Figure 3.25: Pizza predictions

Figure 3.26: Banana

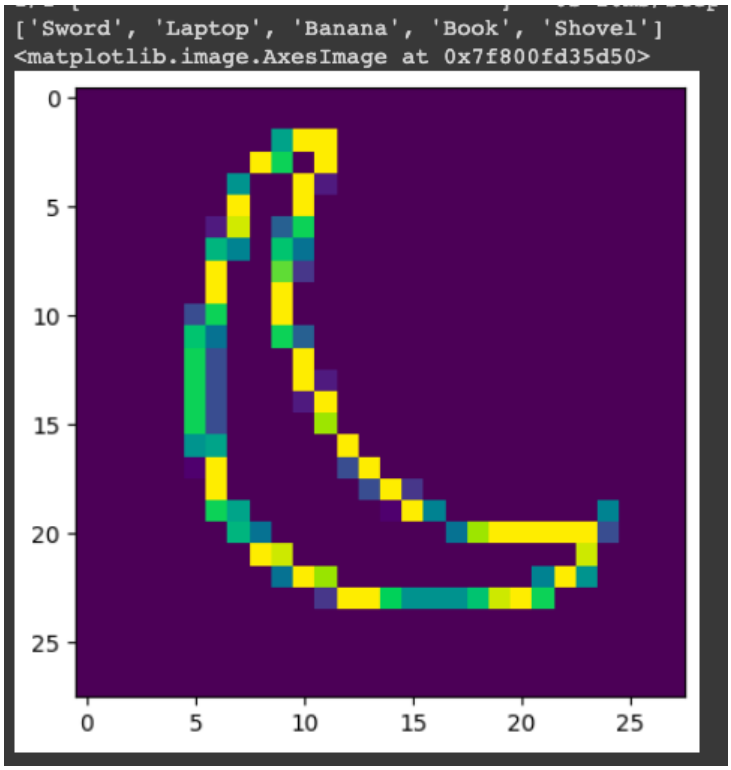

Figure 3.27: Banana predictions

Figure 3.28: Mountain

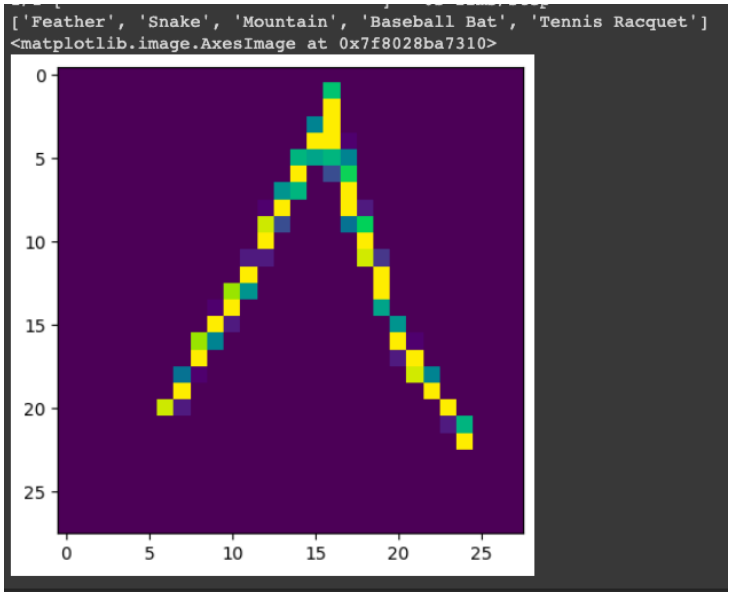

Figure 3.29: Mountain predictions

Figure 3.30: Crown

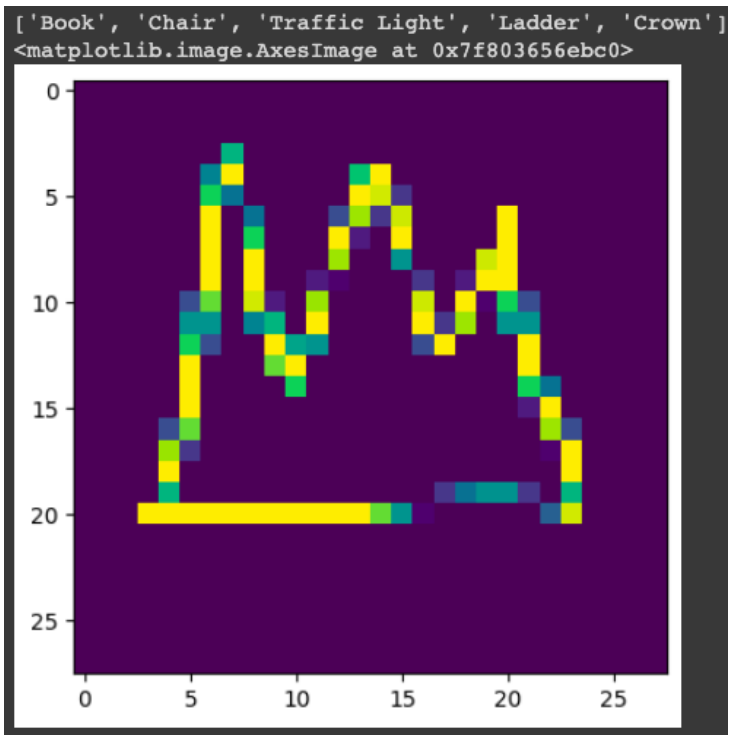

Figure 3.31: Crown predictions

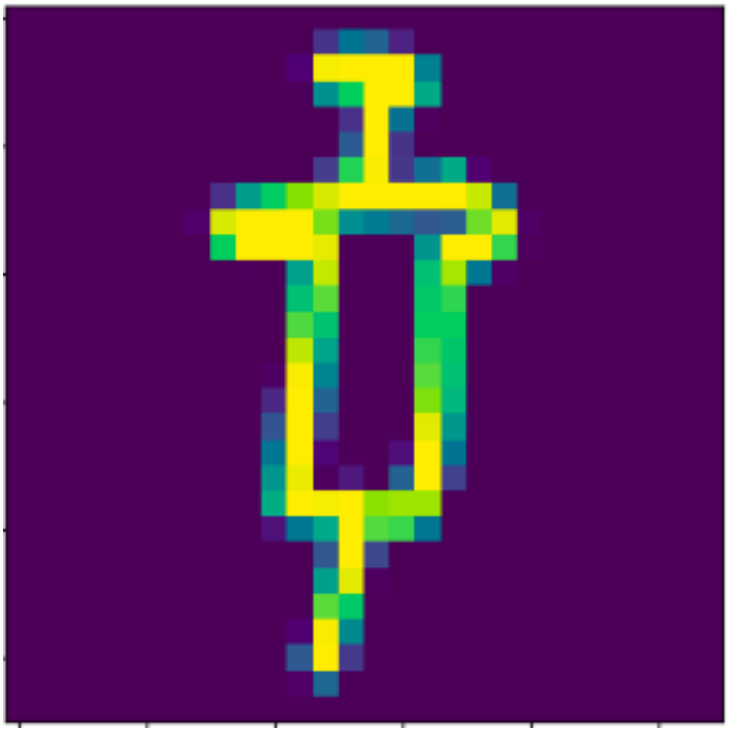

Figure 3.32: Syringe Sketch

```
['Syringe', 'Tree', 'Feather', 'Stop Sign', 'Shovel']
```

Figure 3.33: Syringe Model Predictions

Figure 3.34: Syringe Output Images

Figure 3.35: Tree Output Images

Figure 3.36: Feather Output Images

Figure 3.37: Stop Sign Output Images

Figure 3.38: Shovel Output Images

Figure 3.39: sketch 1

Figure 3.40: sketch 2

Figure 3.41: sketch 3

Chapter 4

Conclusion

In conclusion, Sketch2Search is a promising application of machine learning technology that has the potential to revolutionize the way we search for images on the internet. By allowing users to sketch what they are looking for rather than typing a text-based search query, Sketch2Search provides a more intuitive and accessible search experience. Through the use of convolutional neural networks and other deep learning techniques, Sketch2Search is able to recognize and categorize a wide range of sketches with impressive accuracy. While there is still room for improvement in terms of the application's performance on more complex sketches and its ability to handle a larger dataset, Sketch2Search represents a significant step forward in the field of image recognition and retrieval. As machine learning technology continues to advance, it is likely that we will see even more sophisticated and powerful applications of this kind emerge in the near future.

Chapter 5

Future Work

While the current version of Sketch2Search is capable of recognizing and searching for objects in sketches, there is still room for improvement and further development. Here are some possible avenues for future work:

- **Expansion of object categories:** Currently, Sketch2Search can recognize a limited set of object categories. In the future, more object categories can be added, allowing for a more comprehensive search experience.
- **Improving sketch recognition accuracy:** Sketch2Search's accuracy can be improved by incorporating more advanced machine learning models for sketch recognition.
- **Incorporation of natural language processing:** Sketch2Search can be further improved by integrating natural language processing capabilities. This will allow users to describe objects in their sketches in natural language and have Sketch2Search recognize and search for them accordingly.
- **Creating a user interface:** To further enhance the Sketch2Search experience, a user friendly interface would be needed. It would make it accessible to more people and easier to use.

Overall, there are many potential directions for future work on Sketch2Search that can enhance its capabilities and usability.

Appendix

List of Figures

Bibliography

[1] Analytics Vidhya. Convolutional Neural Networks (CNN). https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/, 2023.

[2] Baeldung. The Dropout Layer in Machine Learning: ReLU Dropout Layers. *Baeldung*, 2023.

[3] Bahadur, Akshay. QuickDraw. https://github.com/akshaybahadur21/QuickDraw, 2022.

[4] Bui, T. et al. Sketching out the details: Sketch-based image retrieval using convolutional neural networks with multi-stage regression. *Computers & Graphics*, 71:77–87, 2018.

[5] Xudong Chen, Xiangyun Gao, and Weijie Li. Scene-based Searching System using Object Recognition and Natural Language Processing. *IEEE Access*, 8, 2020.

[6] Deepchecks. One-Hot Encoding in Machine Learning. https://deepchecks.com/glossary/one-hot-encoding/#:~:text=One%2Dhot%20encoding%20in%20machine,categorical%20data%20in%20machine%20learning., 2019.

[7] Google. Quick, Draw! https://quickdraw.withgoogle.com/, 2016.

[8] Google Creative Lab. Quick, Draw! Dataset. https://github.com/googlecreativelab/quickdraw-dataset, 2016.

[9] Gradio. Building a Pictionary App. https://gradio.app/building-a-pictionary-app/, 2019.

[10] Ibtihaal M. Hameed, Sadiq H. Abdulhussain, Basheera M. Mahmmod, and D T (Reviewing editor) Pham. Content-based image retrieval: A review of recent trends. *Cogent Engineering*, 8(1), 2021.

[11] IBM. Machine Learning. https://www.ibm.com/topics/machine-learning.

[12] Jason Brownlee. Rectified Linear Activation Function for Deep Learning Neural Networks. *Machine Learning Mastery*, 2019.

[13] Łukasz Kobyliński and Krzysztof Walczak. Spatial emerging patterns for scene classification, 2010.

[14] Fang Li, Keren Shi, and Jianxin Wang. Scene-based Searching System using Convolutional Neural Networks. *IEEE Access*, 7, 2019.

[15] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262–282, 2007.

[16] Nihal Mannaa. The Project Link. https://github.com/nihalmannaa/Bachelor-Project.git.

[17] Simplilearn. What is Machine Learning Process? *Simplilearn Blog*, 2023.

[18] Geoffrey Hinton. Yann LeCun, Yoshua Bengio. Deep learning. *Nature*, 521, 2015.

[19] Shih Fu Chang Yong Rui, Thomas S. Huang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10, 1999.

[20] Y. Zhao, B. Hu, Y. Wang, et al. Identification of gastric cancer with convolutional neural networks: a systematic review. *Multimedia Tools and Applications*, 81(11717–11736), 2022.