International Journal of Innovative Research in Engineering
Volume 4, Issue 3(May-June 2023), PP: 492-495.
www.theijire.com **ISSN No: 2582-8746**

Tinker Hunt

Shubham Gulia[1], Anshika Agrawal[2], Harsh Jain[3], Dr. Ankita Gupta[4]
[1,2,3] Department of Computer Science & Engineering, Maharaja Agrasen Institute Of Technology , India.
[4]Supervisor, Department of Computer Science & Engineering, Maharaja Agrasen Institute of Technology, India.

How to cite this paper:
Shubham Gulia[1], Anshika Agrawal[2], Harsh Jain[3], Dr. Ankita Gupta[4], "Tinker Hunt", IJIRE-V4I03-492-495.

https://www.doi.org/10.59256/ijire.20230403117

Abstract: *With tech taking over most of the life process in recent times, primary education still lacks interactive processes using technology in the process of "Learning by Doing". Most of the Ed. techs offer visual learning with no or near-to-zero input by children. Tinker Hunt can be a possible and one of its kind of interactive learning application used to tackle the same. A rough sketching input by children can yield results over theinternet in the form of pictures, detailed information, etc. Tinker Hunt is created using Google's QuickDraw Dataset, using hundreds of sketch patterns for multiple categories of icon-level doodles, collected from hundreds of individuals that are recognized by CNN trained model. Also, the app is tested for high accuracy in sketch recognition and categorization.*

Key Word: *Machine Learning; QuickDraw Dataset; Interactive Learning; Convolutional Neural Network; Artificial Neural Network*

I.INTRODUCTION

Primary education still lacks interactive procedures integrating technology in the process of "Learning byDoing", despite of the fact that technology has taken over the majority of life processes in recent years. The majority of Ed. techs provide visual learning with very little to nearly no child interaction. 'Quick, Draw!' is an online game that Google published in November 2016 that challenges participants to draw a certain item in less than 20 seconds. This isn't a typical game though. After the user draws, a sophisticated neural network makes an effort to determine the object's categorization, and as the user adds more and more information, the predictions it makes change.

Outside of Quick, Draw!, the ability to recognize and classify hand-drawn doodles has significant implications for the development of artificial intelligence in general. For instance, the research in computer vision and pattern recognition, especially in subfields like Optical Character Recognition (OCR), will be greatly impacted by the conception of a robust classifier on high-noise datasets.

For the sake of this project, we decide to concentrate on categorizing the final drawings as a whole. Despite having a simpler goal than the original game, this assignment is nevertheless challenging since there are many categories (10), there are so many different doodles within even one category, and there are so many doodles that are similar that it may be confusing. So, using the Quick Draw! doodle as our input, we build a multi-class classifier whose output is the projected category for the thing being drawn. Since ancient times, drawing has been one of the main methods of communicating ideas.

Young users with limited drawing skills benefit from the informational richness and improved learning overall provided by this participatory strategy. Iterative and interactive sketch-based screen searches are supported by Tinker Hunt. It makes use of a digital drawing board for user input, allowing for live searching and interactive user input through a mouse cursor or touchscreen. All of these work to its benefit to shorten wait times for other search methods.

II.MATERIAL AND METHODS

Hand Drawn Sketch Classification Using Convolutional Neural Networks was written by Habibollah Agh Atabay.[2] The accuracy of sketch image categorization is increased in this research by training a few deep CNNs. Because the input is in the form of small pictures, the architecture of CNN is simplified, allowing it to be trained in a reasonable amount of time and increasing training speed.

Feature-level fusion of deep convolutional neural networks for sketch recognition on smartphones, this paper was developed by the authors E. Boyaci and M. Sert. In this paper, they proposed a feature fusion scheme using AlexNet FC6 and the VGG19 pool 5 layers that achieved better recognition accuracy of 69.175% compared with the standalone CNN models.[3] Free-hand Sketch Recognition Classification, this paper was developed by the author Wayne Lu, Elizabeth Tran. they used a publicly available dataset of 20k sketches across 250 classes. CNN is applied to improve performance toincrease recognition accuracy. Then the effects of several hyperparameters on overall performance were analyzed using a ResNet approach.[4]

Image Classification with Deep Learning and Comparison between Different Convolutional Neural Network Structures using TensorFlow and Keras, this paper was developed by the authors Karan Chauhan and Shrwan Ram. They have useda large number of images of two types of animals - cats and dogs for image classification. 4 different structures of

CNN are compared with 4 different combinations of classifiers and activation functions.[5] Quick, Draw! Doodle Recognition, this paper was developed by the authors Kristine Guo, James WoMa and, Eric Xu. they have used the Quick Draw dataset by Google and have compared two algorithms - KNN and CNN and have achieved an accuracy of 35% and 60% respectively.[5]

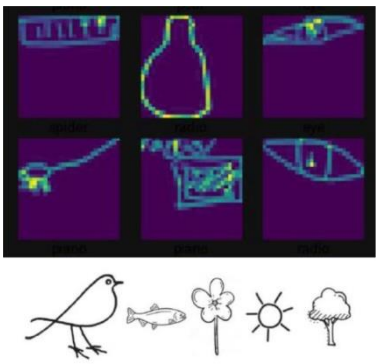

Fig 1. Examples of the doodles available in the Quick Draw Dataset.

A Quick, Draw! dataset with more than 50 million photos divided into 345 categories was made available by Google tothe public. The visuals can be represented in a variety of ways. Each drawing is represented in one dataset as a collection of line vectors, while each image is included in another dataset as a 28x28 grayscale matrix. We utilize the latter versionof the dataset because the goal of this study is the categorization of the complete doodle. Each 28x28 pixel picture is treated as a 784-dimensional vector. We divided the data into three folds to test our models: 15% for validation, 15% for testing and, 70% for training.

By randomly selecting 1% of the drawings from each category, we opted to produce a smaller subset of the original dataset in order to speed up processing and minimize data storage requirements. As a consequence, we get about Ten Thousand (10,000) examples for the testing set and the same number of examples each for the training set. Additionally, each category has a roughly equal amount of drawings, thus the training dataset has about 1000 instances of each type.

Implementation

Figure 2 gives an overview of Tinker Hunt architecture. Tinker Hunt offers a drawing canvas and recognizes any patterndrawn via a deep neural network trained with the QuickDraw dataset. The user lands on the home screen and as soon as they click on the "Get Started" button, a canvas pops up where they can draw or sketch using a touch screen, a mouse, or any interactive input device such as a digital pen. After the sketching is done they can click on the magnifying glass icon to initiate the searching and recognition process. The input is then sent to the backend which is processed using the CNN model and the pattern is then recognized and final prediction result is sent to the backend which then fetches the information from Google using the SERP API. The data such as basic description, images, and more related info is then sent over to the frontend and displayed on the screen. All of this takes place within a very small time interval of nearly 2-3 seconds.

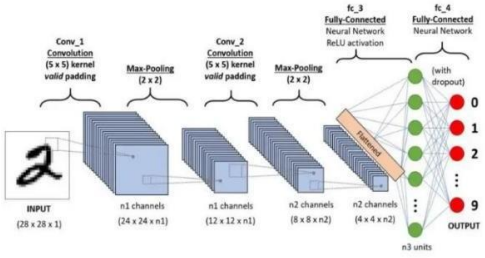

Fig 2. Convolutional Neural Network [6].

A model based on the convolutional neural network deep learning method is used by Tinker Hunt to anticipate the pattern.The Convolutional layer, the pooling layer, and the fully-connected layer are the three foundational layers of every CNN model. The central component of CNN is the convolutional layer, which is also where the majority of computation takes place. It needs input data, a filter, and a feature map, among other things.

Fig 3. Landing Page of Tinker Hunt app

The Kernel or Filter is the component that is present in the initial Convolutional Layer and performs the convolution operation there. K has been chosen as a 3x3x1 matrix. Similar to the Convolutional Layer, the Pooling layer is in the responsibility of reducing the spatial size of the Convolved Feature. The amount of processing resources required to process the data will be decreased through dimensionality reduction. Additionally, by enabling the extraction of dominant traits that are rotational and positional invariant, it contributes to properly training the model.

A 2x2 max-pooling layer has been implemented. In essence, it returns the highest value from the area of the image that the kernel has covered. In order to regularise our neural network and improve results, we have also incorporated a dropout of 0.25. To restructure the data in the previous section, we utilized flatten and dense layers.

Users can draw on Tinker Hunt website via mouse or touch events. Users can choose any pen of their choice to draw. Each time when user clicks on the search button, Tinker Hunt downloads a .png file as well as show appropriate information about the drawing after getting predictions from the backend server and deep learning model. A user We have trained a deep neural network using QuickDraw's network architecture to recognize our drawings. We used a random 1000 samples of each of our 10 QuickDraw classes. We then split our classes into training and test samples and train the network, which yielded an accuracy of 93.34%.

Fig 4. Canvas for sketching and final results for two sketches. The first sketch is recognized as Ice Cream and the second drawing is recognized to be a cat.

III.RESULT

Tinker Hunt is able to anticipate ten different types of drawings. For the model's training, we used 40 epochs and 10,000 data samples for each class. Epochs are represented along the x-axis of Figure as time units, while model accuracy is plotted along the y-axis. An epoch is the total number of iterations required to train the machine learning model using allof the training data at once in a cycle. The underlying model parameters have been updated once throughout an epoch for each sample in the training dataset. An epoch is made up of one or more batches. For instance, a single-batch Epoch is described by the batch gradient descent learning process.

After running all the epochs, the model is ready to predict the name for the input image. To find the accuracy percentage, we loop through the preds array and check if the elements match with the y_test array elements. The code for finding the accuracy percentage is shown in Figure 6. Final Accuracy of the model comes out as 93.34% with 40 epochs.

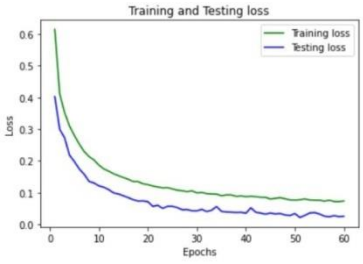

Fig 5. Training and testing loss.

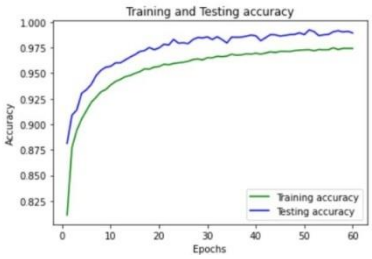

Fig 6. Training and testing accuracy

```
[ ] score = 0
    for i in range(len(preds)):
        if np.argmax(preds[i]) == y_test[i]:
            score += 1

    print ("Accuracy: ", ((score + 0.0) / len(preds)) * 100)

    Accuracy:  93.34
```

Fig 7. Accuracy Percentage

IV.CONCLUSION

The foundation of Tinker Hunt is Google's QuickDraw Dataset, which uses hundreds of sketch patterns for various types of icon-level doodles created by hundreds of people who are identified by a CNN-trained model. It can identify the interactive input drawing and effectively fetch information about it. Tinker Hunt can be a possible and one of its kind of interactive learning application used to tackle the same. A rough sketching input by children can yield them results over the internet in the form of pictures, detailed information etc. In our upcoming work, we'd like to test out advanced CNN architectures like VGG-Net and ResNet, which have already attained cutting-edge levels of image classification performance, but not specifically for sketching. On providing stroke order information and other tract metrics like velocity and acceleration, we anticipate that training our algorithms on the complete dataset will improve the overall accuracy.

We have only used around 1% of the Quick, Draw! dataset overall. Last but not least, we think that ensembling approaches are fascinating, especially for simpler methods like KNN.

References

1. *Ha, D., & Eck, D. (2017). A neural representation of sketch draw- ings. arXiv preprint arXiv:1704.03477.*
2. *Habibollah Agh Atabay, "Hand Drawn Sketch Classification Using Convolutional Neural Networks", Gonbad Kavous University, Iran, 2016.*
3. *E. Boyaci and M. Sert, "Feature-level fusion of deep convolutional neural networks for sketch recognition on smartphones," 2017 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, 2017.*
4. *Wayne Lu, Elizabeth Tran, "Free-hand Sketch Recognition Classification", Stanford University, 2017.*
5. *Kristine Guo, James WoMa, Eric Xu, "Quick, Draw! Doodle Recognition" Stanford University, 2018.*
6. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way.*
7. *Meta Analysis of Deep Learning Models for Doodle Recognition.*
8. *Lu, W., & Tran, E. (2017). Free-hand Sketch Recognition Classifi-cation.*
9. *Rui Hu and John Collomosse. 2013. A performance evaluation of gradient field hog descriptor for sketch based image retrieval. Computer Vision and Image Understanding 117, 7 (2013), 790ś806.*
10. *Forrest Huang, John F. Canny, and Jeffrey Nichols. 2019. Swire: Sketch-based user interface retrieval. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. ACM.*
11. *Alex Krizhevsky, Ilya Sutskever, and Geoffrey E.Hinton. 2012. ImageNet classification with deep convolutional neural networks. In Proc. 26th Annual Conference on Neural Information Processing Systems (NIPS). NIPS, 1106ś1114.*
12. *Jonas Jongejan, Henry Rowley, Takashi Kawashima, Jongmin Kim, and Nick Fox-Gieg. 2016. Quick, Draw! https://quickdraw.withgoogle.com/ Accessed March 2022.*