Swift Draw : Real-time Doodle identification using CNN

Abhilash Thankachan[1], Neevan R[2]
Department of Master Of Computer Applications Engineering
[1] College Of Engineering, Trivandrum, Kerala, India
[2] Assistant Professor, College Of Engineering, Trivandrum, Kerala, India

Abstract- *Hand-drawn doodle recognition has significant importance in the real world. It is very difficult and hidius task to recognise the hand-drawn doodles.In this paper I use convolution neural network to recognise the doodle drawn by the user. User can draw on the interface with the help of webcam. User draw the image to canvas by simply moving the finger or an object in front of the webcam. The quick draw system will track the movement of the object in webcam and draw on the interface according to the movement.The setup consist of a single camera to track various object movements by the user and take this movements as an input to the system. The system will analyze the doodle drawn and guess what the doodle represent using the convolutional neural network. The application will replace the user drawn image with the previously drawn image if the prediction is correct like the Google AutoDraw do.*

Keywords- Convolutional Neural Network(CNN),Doodle.

I. INTRODUCTION

“Swift Draw“ is a paper that has been developed with the ideas inspired from two Google projects ‘Quick, Draw’ and ‘Auto Draw’.Quick!Draw is a Google online game ,released on November 2016.The aim of the game is challenging the players to draw the given object within the provided 20 seconds. the twist with the game is that ,here advanced neural network is used to guess the category of the object, as the player adds more and more details. On the other-hand, AutoDraw freely allows the user to draw what they wish and at the same time Google's algorithm will try to find out what you wish to draw and recommend clip art on the top bar. By selecting the suggested clip art your drawing will be replaced with the selected one. But the hardest point with both these techniques is that interactivity with the softwares are through hardware pointing devices or a touch sensitive screen.

So this paper aims to club the drawing techniques, AutoDraw and Quick!Draw with a more attractive input device-”Human Hand!!”.Additionally a pen or any pointed device can also be used as an input device. For this camera based Human Computer Interaction(HCI) is exercised.The advantage with HCI are:

- Workspace used : The system uses a camera for interacting ,thus additional workspace are not required.
- Stripped Installation: No external hardware and software requirements are needed for the system users.
- Atrractive user intraction: User can change the direction and speed of the hand movement ,which makes the system easy to use

II. METHODOLOGY

Swift draw consist of two phase – ‘Drawing’ and ‘Predicting’ sections. Drawing phase is the work space were user interacts with system. Initially, record the hand movements. Later background subtraction is done to acquire mono color palm. In case, if a pen is used as input device then the mono color pen is received after subtraction. Next step is to contour the palm/input device. When the contour image of the palm/device is obtained then the edges are detected. Finally the last section, the drawing is done. The user’s finger movements or the input device movement is recognized and is then converted into drawings which is finally saved.

Next phase is the Prediction. In prediction section ,the saved image is inputed to CNN model and the prediction is done and the similar images are shown to the user.

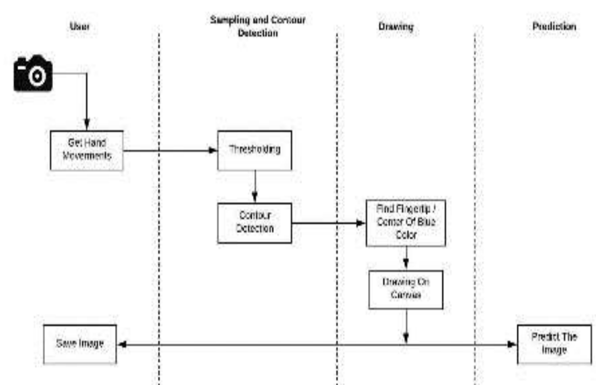


Fig 1: Overall design.

Convolution Neural Network

A state-of-the-art model that is well known for its ablility to recognize and quickly learn local features within an image is the Convolutional Nueral Network.The model architecture used in this system is shown below.

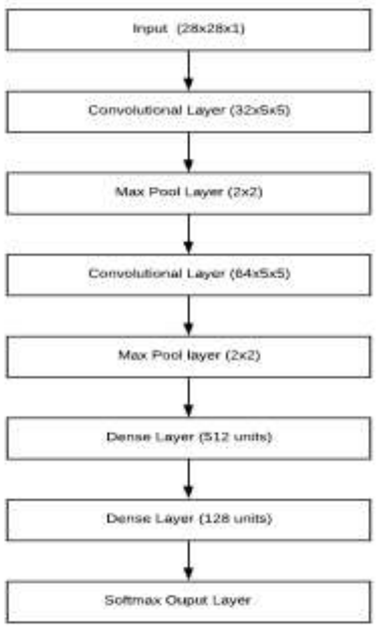


Fig 2: Layers used in CNN

A numpy bitmap file (each numpy file consist of 28x28 grayscale bitmap of hand drawn doodles) is passed to the convolutional filters of size 32x35x5.The output then goes to a max pooling layer with a kernel size of 2 × 2.The output from the max pool layer goes to the next convolutional filters followed by the max pool layer. At last,the result is passed through three fully-connected or dense layers. Dropout and ReLu activation function are also used with each layers.The output then moves to one more affine modification to get logits of dimension 300 (number of categories) before the softmax is applied to obtain probabilities for each class.
Cross-entropy loss is used to train the model.

III. IMPLEMENTATION

The Swift Draw system has two core modules, Real-Time Drawing, and Doodle Prediction. The first module allows the user to draw doodle in canvas in real time with the use of a single web camera. Users can use either a blue object or his/her bare fingertip for drawing the doodle. In the doodle prediction module, the CNN model for doodle recognition is created and it is used for the doodle classification.

A. Real-Time Drawing

For capturing the movements of the user, the system uses the OpenCV videocapture() method. Video is grabbed frame by frame using a recursive function call. In this case, we pass 0 to the videocapture() method to read from a webcam. Once the system started to get the webcam feed, the system will constantly look for a blue color object in the frames with the help of the cv2.inRange() method using the blue color HSV range. A series of image operations like erosion and dilation to make the Region Of Interest(ROI) smooth and clear(here ROI is blue color) is also done by the system. The system finds out the contour of the ROI. The Swift Draw uses the center of ROI as the drawing brush. Then the system tracks the coordinates of each and every point of the center of the contour, as it moves on the screen and stores these coordinates in a deque. The deque data type is used for the fast insertion and deletion of points. Now the system just joins the points in the deque using a line in the canvas with the help of the OpenCV function cv2.line(). The system joins all the points in each and every frame with a line, place it in the canvas created using numpy array. Once the drawing is done, the drawn doodle is passed to the prediction module on users affirmation. For drawing with fingertip, the user has to show his hand to the camera and the system take skin color samples. These skin color is used for background subtraction and finding Region Of Interest (here region of interest is hand). After finding ROI, the farthest point from the center is considered as the pointed fingertip and it is used for drawing.

When it comes to image classification, there is no better choice than a Convolutional neural network. The classification model is created using CNN. There are two convolution layers which perform the feature extraction. The first layer considers thirty-two features and the second use sixty-four features. Both these layers use ‘ReLu’ as theactivation function. In order to reduce the size of the images, we are using a max-pooling layer along with each convolution layer. For training, we are using two layers with 512 and 128 nodes respectively. The last layer of the model is the output layer. The output layer uses the softmax function as its activation function. It classifies a given input into one of the 300 predefined classes

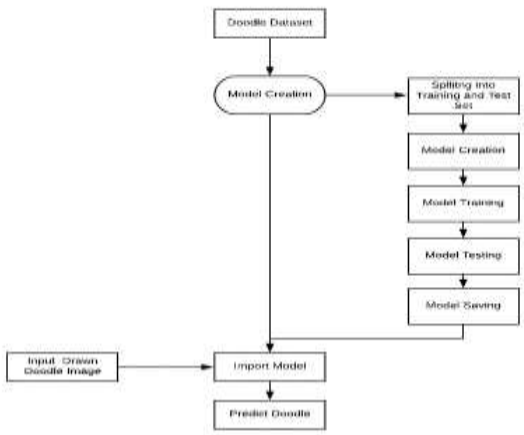
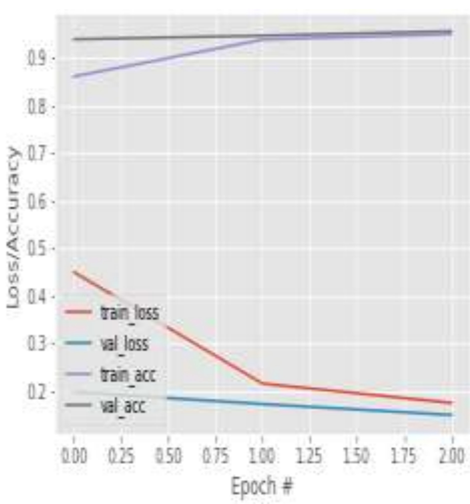


Fig 3: Model working.

IV. RESULT AND INFERENCES

To achieve best performance by the CNN model several parameters in the model skeleton such as dropout rate,number of units in each dense layer and number epochs are adjusted. The best performance is achieved with 2 dense layers with 512 and 128 units in each layer.Each layer has a drop out rate of 0.6. Its tested and validated that the convolutional neural network model with images from the dataset and the model has higher accuracy rate of 98.38 percentage



The swift draw system is tested with user drawn doodle and classified the doodle to accurate classes.The model in real world scenario has lower accuracy rate compared to the model tested with dataset images.

V. CONCLUSIONS

The swift draw system allows user to draw doodle in real-time with the help of a single web camera and the system identifies the category of the doodle. On the affirmation of the user on the predicted doodle user will get list of images as suggestion similar to doodle drawn by the user.The selected image will replace the doodle drawn by the user.There is no other system is available that allows the user to draw with bare hand.

This project is inspired by two renowned projects of Google namely Quickdraw and Auto draw. The major motive behind this work is to provide a platform to draw pictures for those who are not gifted with the ability to draw pictures. The aim is to develop an intelligent drawing software which aids such people to draw pictures. While the user starts doodling, the intelligent system will give suggestions, along with the suggestions it will be displaying a collection of perfectly drawn images as well. The user can choose an image that best meets their needs. Then the user can draw the next element and this process continues for every picture drawn by the user. In such a way the user can draw multiple images to complete their picture. In essence, the future extension of this work would be an intelligent drawing guide

Various segmentation techniques such as p-tile thresholding, watershed segmentation can be applied to increase the accuracy of the cancer detection system. Some feature set along with morphological features can play a positive impact on accuracy. A new class can be added to the system for prediction of the possibility of lung cancer if more dataset is available.

REFERENCES

[1] Kristine Guo, James WoMa, Eric Xu ,"Quick, Draw! Doodle Recognition", Stanford University.

[2] Reema Arora, Renu Kumari, Shabnam Kumari ,"Real Time System Controlling Using A Web Camera Based On Colour Detection", IJESRT International Journal Of Engineering Sciences Research Technology.

[3] Towards Data Science (2019). Doodling with Deep Learning!. [online] Available at: https://towardsdatascience.com/doodling-with-deep-learning-1b0e11b858aa [Accessed 9 May 2019]