Doodle classification using convolutional neural network

Aniisa Bihi[1], Amanda Tydén[2]

Abstract
To create a doodle classifier, inspiration was drawn from Google's experiment with artificial intelligence called "Quick, Draw!". The goal was to create a game where the user was presented with a word to draw, the implemented model would then guess the word. If the model would guess the doodle correctly then a new word would appear for the user to draw. To reach this goal a training model was created using Google's data set and TensorFlows tutorial for recurrent Neural Networks. Only 100 categories out of 345 were selected from the data set and 6000 images per class were trained and tested for the model. The model was created out of three convolutional layers and two dense layers. During the training phase it showed good performance and when further implemented into a user interface it showed great results. Depending on the user's doodle and that user's perception of the object presented, the model will perform differently. In the end it became an abstract question on how one object actually looks and how the majority of the population choose to represent them when faced with the task to draw that object.

Source code: https://github.com/Tyddan/TNM095-AI-doodle

Website: https://tyddan.github.io/TNM095-AI-doodle/

Authors
[1]*Media Technology Student at Linköping University, anibi335@student.liu.se*
[2]*Media Technology Student at Linköping University, amaty364@student.liu.se*

Keywords: AI (Artificial Intelligence) — Neural Network — TensorFlow — Keras — Colab (Google Colaboratory)— ML (Machine Learning)— CNN (Convolutional neural network)

Contents

1. Introduction

In computer technologies, image recognition is known as a technology that uses algorithms and machine learning concepts to classify and therefore recognize images. With machine learning it is possible to train computers to "see" and process information like humans, thus calling it artificial intelligence. [1]

Deep learning is a popular subset of machine learning methods based on artificial neural networks. It is based on several deep learning architectures such as convolutional neural networks. CNN is mostly known as an image classification algorithm and can therefore be used to build a classification model for such purpose [2]. An example of this is Google's experiment with neural networks, called "Quick, Draw!", where the inspiration for this project is drawn from.

Google's Quick, Draw! is an online game using neural network to classify the users doodles [3]. The data set used is a continuing expanding data set containing 345 different classes with more than fifty million images of drawn doodles [4]. The data set expands and the AI learns every time the game is being played. By saving the doodles drawn by each and every player, the game only gets smarter. The data set is shared publicly to help with machine learning research and has been used in this project while building a classification model.

To build a similar game like Google's with an appropriate classification model, TensorFlow and Keras can be used. TensorFlow is an end-to-end open source machine learning platform. The library is used to develop and train machine learning models [5]. Together with the neural networks API Keras, it is possible to build and train ML models easily. Keras' data structure model can be used to organize layers, simply by creating a linear stack of layers called a sequential model. [6]

Based on Google's Quick, Draw!, TensorFlow and Keras' cooperative methods, a model was created for this project. The goal was to understand TensorFlow and convolutional neural networks better by creating a doodle classification game.

2. Theory

2.1 Neural Network

Deep learning is a subset of machine learning and it is built using neural networks. Neural networks takes in inputs, which are then processed in hidden layers using weights that are adjusted during training, the model then spits out a prediction. The weights are adjusted to find patterns in order to make better predictions. The user does not need to specify what pattern to look for, the neural network learns on its own. [7]

2.2 CNN for image classification

The main task for image classification is the definition of the input images class. This skill is something people learn from their birth but computers see images different from people. The computer looks for the characteristics of the base level. When training the model, the prediction can only be as good as the images it trains on. If the data set is too small for the training phase the test images need to be very similar to the training images. The more varied data set the better prediction. An aspect to also look at when talking about how people draw is the psychological. Depending on where in the world a person comes from the perception of an object can be different, for example a door can look different in different countries. [7]

2.3 Convolution layer

Computers see images using pixels, a certain group of pixels may signify an edge in an image or some other pattern. Convolutions use this to help classify images. The first layer in a CNN is always a convolution layer. In the convolution layer a filter slides over all pixels in the image. As the filter slides, or convolutes, over the input image it multiplies all the values in the filter with the original values of the image [8]. This process is repeated for all images in the training data resulting in a new array of numbers resulting in a feature map.

2.4 Pooling layer

Max pooling is a pooling operation that calculates the maximum, or the largest, values in each patch of each feature map. Max pooling is often performed after the convolution layer. The results are down sampled, or pooled, feature maps that highlight the most present feature in the patch, not the average presence of the feature that is the case of average pooling. This has been found to work better in practice than average pooling for computer vision tasks like classification. [9]

2.5 Dense layer

The dense layer represents a matrix vector multiplication. The values in the matrix are the trainable parameters which get updated during back-propagation. All nodes in the previous layer connect to the nodes in the current layer. Increasing the number of nodes in each layer will increase the model capacity. It introduces non-linearity that does not affect the receptive fields of the convolution layer. Mathematically this means that the dense layer applies rotation, scaling and translation transform to the vector. It is in the dense layer that the classification takes place. [10]

2.6 Dropout layer

The dropout layer is a technique where randomly selected neurons are being ignored during training. The purpose of this is to make the network less sensitive to specific weights of neurons. With the hope of making the network capable of better generalization and less likely to over fit. [10]

2.7 Optimizer

The optimizer controls the learning rate. Adam is an optimizer which is generally good for many cases. The Adam optimizer adjusts the learning rate throughout training. The learning rate determines how fast the optimal weights for the model are calculated. A smaller learning rate may lead to more accurate weights, but the time it takes to compute the weights will be longer. [11]

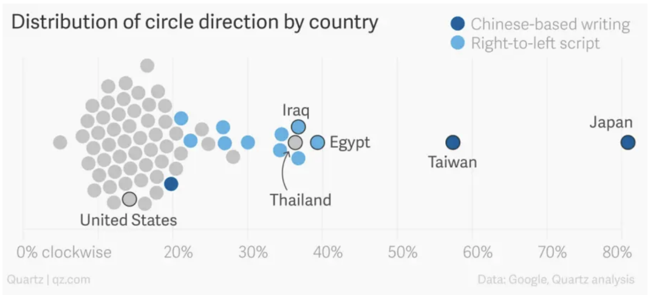

Figure 1. Distribution of circle direction by country. [13]

2.8 Human factor

When creating a neural network the human factor has to be taken into deeper consideration. Can a CNN model actually learn and understand everything? Will it be able to classify any hand drawn image by any person in the world? Research show that there are patterns found in how people from the same country draw different objects. The same object can be drawn in different shapes and sizes all over the world and depends on where in the world you are located and what references you have from your environment. [12]

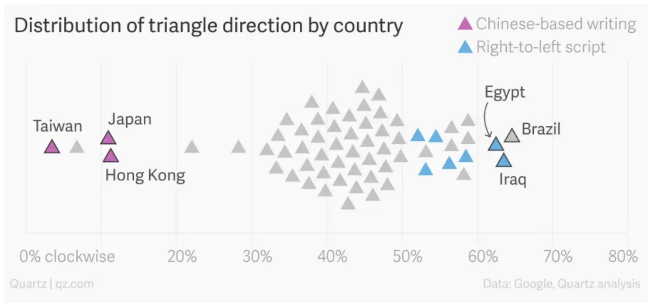

Figure 2. Distribution of triangle direction by country. [13]

A study based on how people around the world draw a circle showed that it could be connected to languages. The way people write - left to right or right to left - could be connected to whether they draw a circle counterclockwise or clockwise, see Figure 1. Even stronger differences were found looking at the direction people draw triangles, see Figure 2. [13]

Other studies showed that more naturally recurring objects looked similar across the world. These objects were for example cats, trees, rainbows or skulls. Figure 3 shows the similarity between four countries drawing a cat, with several drawings layered on top of each other to show a pattern for each country. Objects that showed notable differences between countries were for example sandwiches, mugs and chairs. With these objects, perspective and preferences played a big role in how the objects were drawn. Figure 4. shows how a chair was drawn in different countries, where they were drawn both facing sideways or forwards. [12]

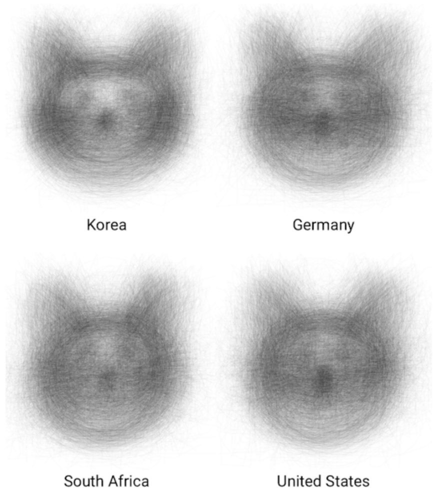

Figure 3. Pattern of cat doodles in four countries across the world. [12]

Looking at the patterns across the world it is easy for a neural network to classify a class by the most common drawing style in the world. This contributes to the neural network only recognizing a class by one style and therefore having a harder time classifying the same class drawn in a less common style. [12]

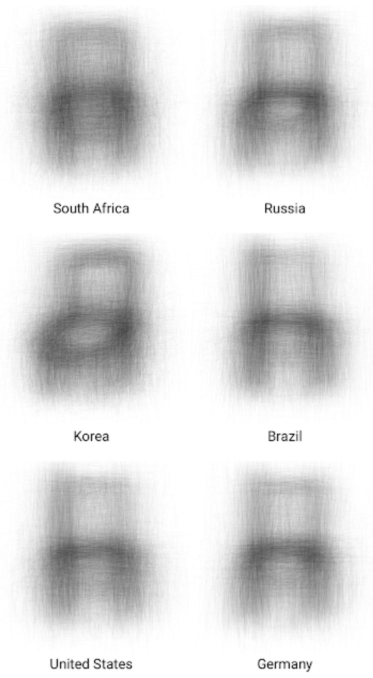

Figure 4. Pattern of chair doodles in six countries across the world. [12]

3. Method

To train and build the model before use, Google Colaboratory was used. Colaboratory is a free open-source development environment that runs completely in the cloud. By using Colaboratory the model could be trained on free processing power while using Keras. TensorFlows JavaScript would then be used to run the model on the browser. [14]

The data set was downloaded directly into Google Colaboratory from Google's "Quick, Draw!" database. 100 classes were selected out of 345, to still have a range of classes but not make it too heavy to calculate. The database contains over 50 million images but was restrained to 600 000 images, thus 6000 images per class. The images were preprocessed and reshaped to 28x28 pixels and split into training and test sets.

The model was created using Keras' sequential model. The model was built up with three convolution layers combined with three max pooling layers and two dense layers. The activation for the layers is rectified linear activation since it is proven to work well with image classification. After the model was built, the training phase followed using TensorFlows Adam optimizer. The model was trained on the images in five epochs and 256 batches with 10 % validation split. After each epoch the model improved itself. After this an evaluation of the training was executed to measure how well the model performs. The model was then saved in a JSON format and downloaded to be implemented in a web interface.

To implement the JSON formatted model in the interface, TensorFlows own JavaScript library was used.

To present the model with input images on a web interface, the input images were scaled and cropped to match the size of the images used during training. The input images are then sent to the model which returns the class with the highest prediction probability.

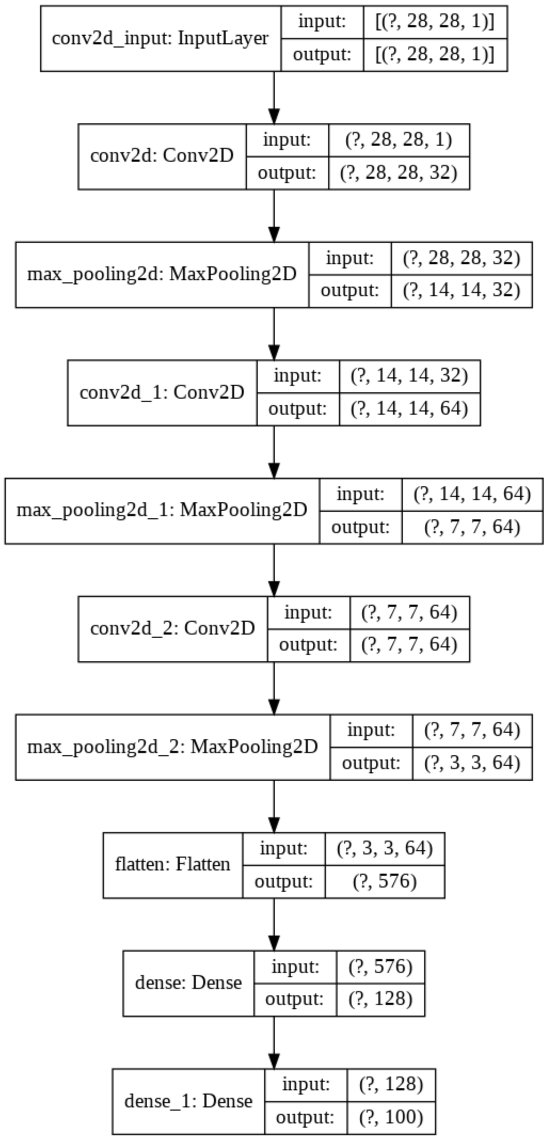

Figure 5. Architecture of the final model.

4. Result

After the model was trained, tested and evaluated, the models test accuracy reached a percentage of 93.85 %. The configuration of the model resulted in a JSON format that could recreate and initialize the same model. The model's architecture can be seen in Figure 5.

After preparing the model for web format, an interface was created. The interface can be seen in Figure 6 and includes a canvas for the user to doodle on. The user is presented a word randomly from the 100 classes that it should doodle while the model guesses. The models guesses and the prediction probability of those guesses are shown in a speech bubble to the right. While the model guesses the doodle, the user has the option to clear the canvas or pick a new word to doodle.

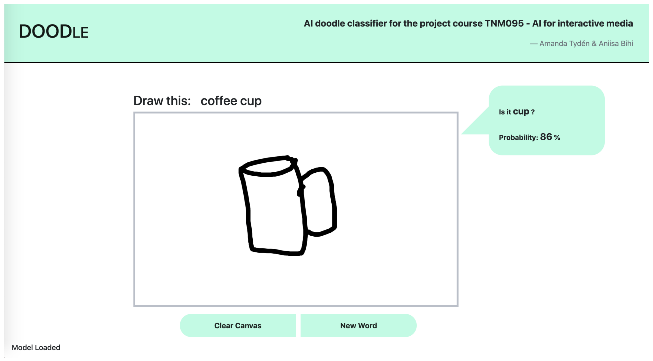
DOODLE
AI doodle classifier for the project course TNM095 - AI for interactive media
— Amanda Tydén & Aniisa Bihi
Draw this: coffee cup
Is it cup ?
Probability: 86 %
Clear Canvas
New Word
Model Loaded

Figure 6. The model presented in a web format.

If the model is able to guess the users doodle correct, as illustrated in Figure 7, a snackbar with the text "AI guessed correct!" will appear. The interface will then reset by clearing the canvas and presenting the user with a new word to doodle.

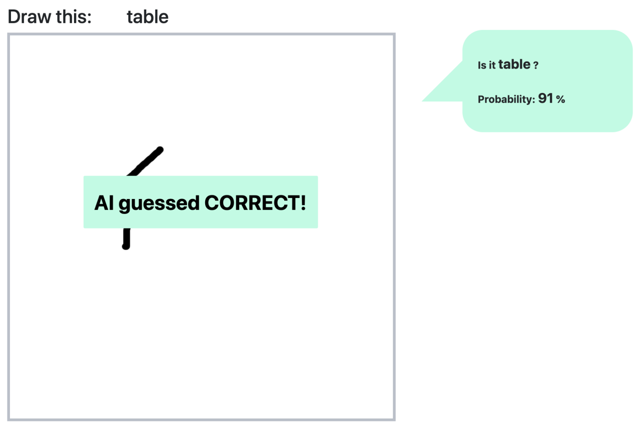

Figure 7. View when the model guesses the doodle successfully.

5. Discussion

The model has a high accuracy according to the result of the evaluation. This result is received using the test set from the "Quick, draw!" data set. The accuracy will be higher for this data set since all the image in the database has been preprocessed equally. The input image drawn in the interface is also preprocessed but there is no guarantee that it is done in the same way. The size of the brush used to doodle may also impact the performance. This is because the image is resized and scaled. The brush size can therefore be too thin to match the brush size of the training data. The CNN model may therefore have difficulties finding a match. If the canvas is large the brush size needs to be thicker otherwise the strokes will be to thin when sent into the model. There were some experimentation on the brush size and it landed on a thicker size because the performance increased. One way to test the real accuracy performance of the model could be to use a different data set and compare the positive/false positive cases and from that see how the model is performing.

Depending on how the model is implemented it will perform differently. More or different layers will affect the performance. One layer that was experimented with was the dropout layer which is supposed to prevent over fitting of the model. When implemented the result was even worse and in the end the model did not have a dropout layer. The pooling layer also prevents over fitting and that might be the reason the dropout layer is not needed for this model.

One thing that could be done is to experiment with different number of convolutional layers to find a better model. Since the implementation presented is based on other similar work it has proven to work well for this kind of classification.

There is also the aspect of the human factor that can be taken into account. Even if the database used is large and international, there can be a lack of variety in the data. Classes that are not as simple and have a large number of drawn varieties, can lose a lot of its data as the neural network learns to recognize the most frequently used style. For example, shoes come in different shapes and sizes and can therefore be drawn differently depending on a person's perspective. If one style, for example sneakers, is highly represented in the large data set, then that is the style the neural network will recognize. The neural network should be trained so that it can recognize all styles of a class, since one style does not fit all.

There are different methods that can be used to improve the variety of data. The protocol for the content generation could be changed so that it focuses on one specific group at a time. In that way the data can represent more accurately a local or global population. Another way to improve the data would be to augment and re-weight it so that the data is more inclusive.

6. Conclusion

Convolutional Neural Networks work well with image classification. One interesting thing to look at is the perception of how a users doodles a specific object, that can depend on the perspective and the geographical location of the user. Since the projects model resulted in a high prediction accuracy it can be concluded that image classification can be done with three convolutional layers, three max pooling layers and two dense layers. Generally speaking there are a lot of different ways the model could be improved, and to focus on that aspect alone would be a project of its own.

References

[1] Ming Liang, Xiaolin Hu; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3367-3375

[2] Saha S. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. 2018. [Accessed 2019-10-17]. Available from: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[3] Google Creative Lab. Quick Draw! 2017. [Accessed 2019-10-17] Available from: https://experiments.withgoogle.com/quick-draw

[4] Google Creative Lab. The Quick, Draw! dataset. 2019. [Accessed 2019-10-17] Available from: https://github.com/googlecreativelab/quickdraw-dataset

[5] TensorFlow. [Accessed 2019-10-17] Available from: https://www.tensorflow.org

[6] Keras. Keras: The Python Deep Learning library. [Accessed 2019-10-17] Available from: https://keras.io

[7] Sorokina K. Image Classification with Convolutional Neural Networks. 2017. [Accessed 2019-10-17] Available from: https://medium.com/@ksusorokina/image-classification-with-convolutional-neural-networks-496815db12a8

[8] Deshpande A. A Beginner's Guide To Understanding Convolutional Neural Networks. 2016 [accessed 2019-10-17]. Available from: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

[9] Brownlee J. A Gentle Introduction to Pooling Layers for Convolutional Neural Networks. 2019. [Accessed 2019-10-17] Available from: https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/

[10] Yang Yang, Lei Zheng, Jiawei Zhang, Qingcai Cui, Zhoujun Li, Philip S. Yu; TI-CNN: Convolutional Neural Networks for Fake News Detection, 2018, Cornell University.

[11] Khandelwal R. Overview of different Optimizers for neural networks. 2019. [Accessed 2019-10-17] Available from: https://medium.com/datadriveninvestor/overview-of-different-optimizers-for-neural-networks-e0ed119440c3

[12] Jana R. Exploring and Visualizing an Open Global Dataset. 2017. [Accessed 2019-10-17] Available from: https://ai.googleblog.com/2017/08/exploring-and-visualizing-open-global.html

[13] Ha T-H. Sonnad N. How do you draw a circle? We analyzed 100,000 drawings to show how culture shapes our instincts. 2017. [Accessed 2019-10-17] Available from: https://qz.com/994486/the-way-you-draw-circles-says-a-lot-about-you/

[14] Google Colaboratory. 2019. [Accessed 2019-10-17] Available from: https://colab.research.google.com/notebooks/