Classification Models for Sketch Drawings

Aparajita Dutta[1], Sarah Saki Robinson[2]

[1]Department of Electrical Engineering, Stanford University
[2]Department of Computer Science, Stanford University

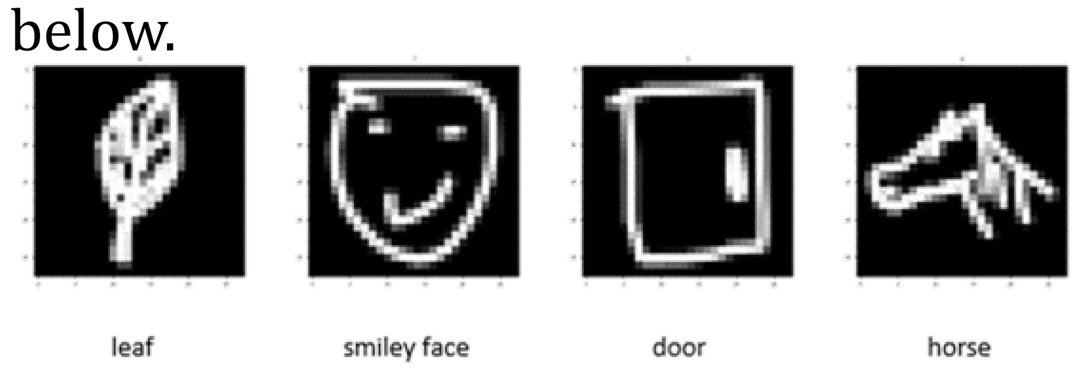
ABSTRACT

Numerous papers have been published, analyzing various algorithms for image classification on photographs. Majority of publications on image classification perform experiments on pictures where the objects in images are easily identifiable. Here, we explore a case where the images of objects are not as easily identifiable by humans.

We apply different classification techniques to a dataset of sketch drawings to find what model performs best.

OBJECTIVE

- Take a hand-drawn image of an object and accurately identify it.
- Examples of the compressed image inputs and their respective outputs are shown below.



DATASET

- Used the Google QuickDraw dataset[1].
- Images of simplified drawings, centered and formatted to a 28 x 28 grayscale bitmap.
- A small subset containing 22 objects was chosen for this project; namely: Apple, Banana, Bicycle, Birthday Cake, Butterfly, Candle, Computer, Door, Drums, Firetruck, Hat, Horse, Ice Cream, Leaf, Panda, Peanut, Pencil, Rainbow, Smiley Face, Snowman, Soccer Ball and Umbrella.
- Data was then split in a 60:20:20 ratio into the training, development and test sets.

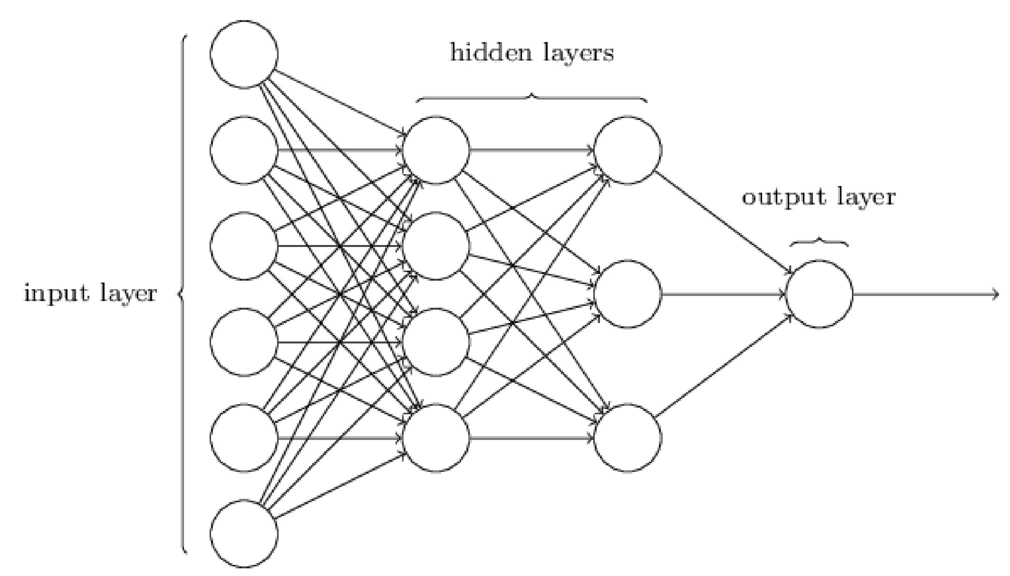
METHODS

Multi-class Logistic Regression

- Generalization of Logistic Regression
- Sigmoid logistic function is replaced by the SoftMax function

Multi-Layer Perceptron (MLP)

- Simple Neural Network
- Our model: 2 fully-connected hidden layers with ReLU activation and an output layer with SoftMax activation

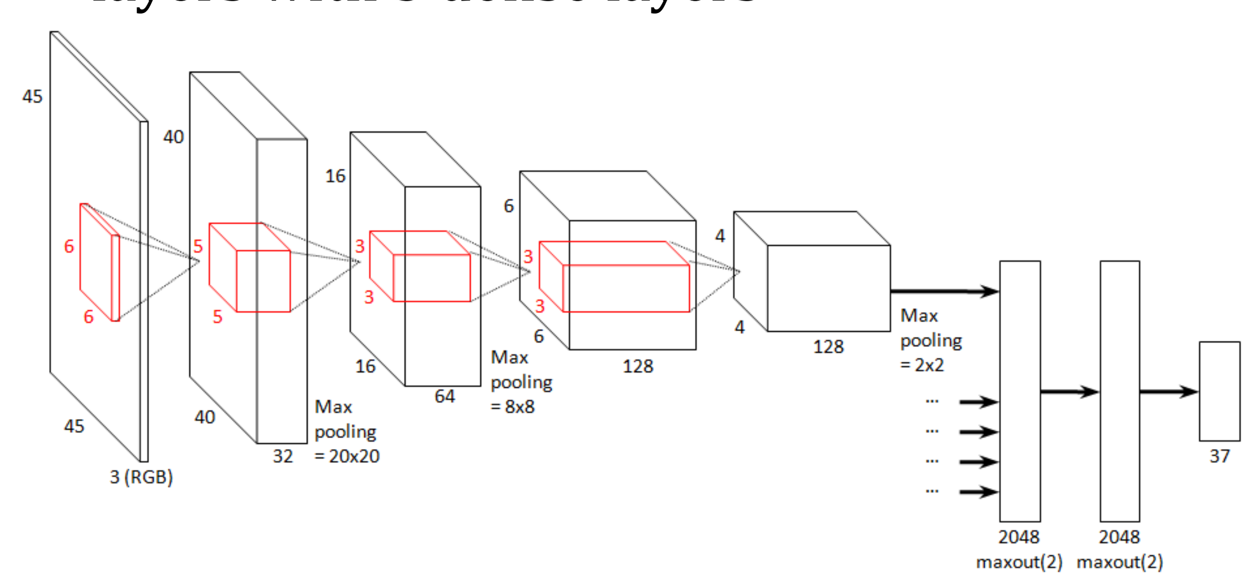


Naïve Bayes

- Based on applying Bayes Theorem
- Assumes conditional independence between feature pairs

Convolutional Neural Network (CNN)

- Regularized version of MLP
- 2 different models: 3 convolutional layers with one dense layer and 2 convolutional layers with 3 dense layers



RESULTS

- Baseline: Majority classifier. Log loss = 33.003, Accuracy 4.447%
- Oracle: Human classifier. Log loss = 1.099, Accuracy = 96.818%
- Method comparison: Performed both general and model-wise hyperparameter tuning to pick the best model for each method. The 4 methods were applied on the validation set. Results are compared below.

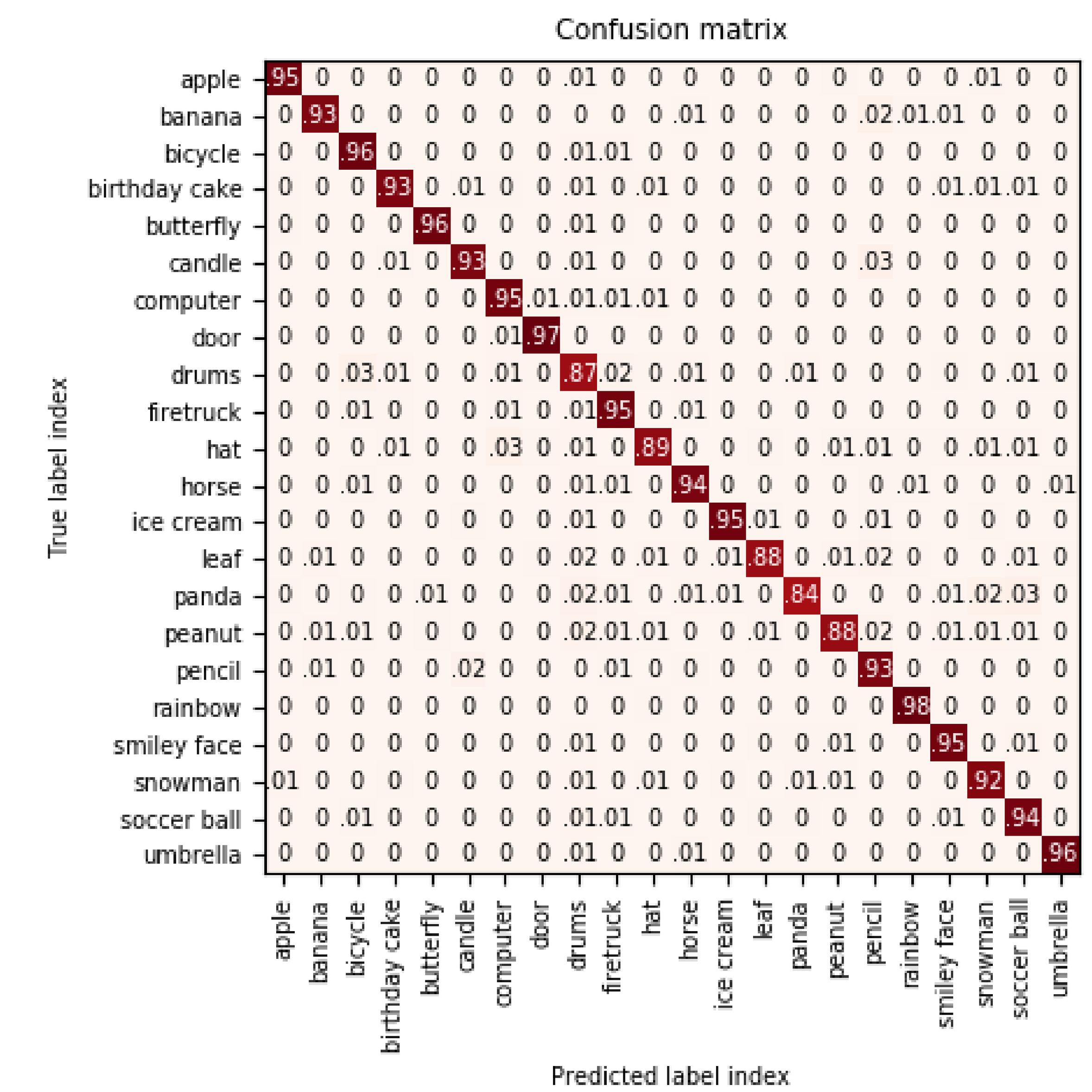
Method	Train Loss	Test Loss	Accuracy	Precision	Recall
Logistic Reg	9.9290	10.2584	0.7030	0.70	0.70
Naïve Bayes	15.6298	15.5872	0.5487	0.56	0.55
MLP	4.5148	5.2632	0.8476	0.86	0.85
CNN	2.1749	2.3804	0.9311	0.93	0.93

CNN performed the best on the validation set. On the test set, it achieved log loss = 2.414 and 93.01% accuracy!

The demo application uses this CNN model to try and accurately classify the doodle input from the user. The performance of the demo app is much lower than the results above. This is due to the difference in data format between the training data and user input.

RESULTS (continued)

Below is the confusion matrix for the CNN model's performance on the test set.



CONCLUSION

- Deep learning methods performed better than traditional methods in most cases because they can learn more about the drawings, with the use of hidden layers.
- CNN performs very well on image data as it can extract information from smaller parts of the image.

Note: All methods were implemented using Scikit-learn[2] and Keras[3].

REFERENCES

[1] Jongejan, J., Rowley, H., Kawashima, T., Kim, J. & Fox-Gieg, N. (2016) The Quick, Draw! - A.I. Experiment. https://quickdraw.withgoogle.com/data

[2] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011) Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, vol. 12, pp. 2825–2830.

[3] Chollet, F. (2015) Keras.https://keras.io.

MLP and CNN images: https://www.peculiar-coding-endeavours.com/assets/