Logo classification Using Modified Pretrained Machine Learning Models

Arjun Singh
Department of Computer and Information Science and Engineering
University of Florida
Gainesville, Florida
singha@ufl.edu

Raghuveer Chandrashekhar
Department of Occupational Therapy
University of Florida
Gainesville Florida
rchandrashekhar@ufl.edu

Venkata Satya Sai Subhash Vadlamani
Department of Computer and Information Science and Engineering
University of Florida
Gainesville, Florida
v.vadlamani@ufl.edu

Abstract: *Over the past two decades, there has been a large rise in piracy, phishing, and copyright infringement crimes involving the use of fake brand logo images. Image classification using machine learning (ML) algorithms provides a solution to address this problem. While there are multiple ML algorithms used for logo classification, the advent of deep learning algorithms using convolutional neural networks (CNNs) has become the standard approach for image classification problems due to their high efficiency and accuracy. In this project, we aimed to develop and apply a multi-layer CNN for the task of image classification in a dataset of 10 different brand logos and the results show promise in identifying the different brand logos.*

Keywords—Brand logos, logo classification, Image classification, Convolutional neural networks, Deep Learning

I. INTRODUCTION

Over the past two decades, with the advances in the field of machine learning and/or natural language processing, there has been a significant rise in cybercrimes such as piracy, phishing, identity theft, copyrights/intellectual property infringement, etc.[1], [2] One common method employed in such cybercrimes involves using images of fake brand logos to trick individuals into divulging personal information such as bank account details and social security numbers.[1] Brand logos can include distinctive features such as text, images, symbols, shapes, and/or a combination of these features.[3] Since all brand logos are unique and there is no standard template or clear definition of what a logo is or what it should contain, it makes it harder to distinguish the fake from the real logos. One way to address this problem is to develop algorithms for image classification using machine learning.[4], [5] Different "traditional" approaches used in logo detection and classification such as logistic regression, K-means, K-Nearest Neighbors clustering (KNN), and support vector machines (SVMs) have been extensively studied in the past.[6], [7] However, the advent of deep learning techniques such as convolutional neural networks (CNNs) have revolutionized image classification as CNNs have been reported to significantly improve the accuracy and efficiency of image classification. [4], [8] This is because CNNs, especially those with multiple layers, have more powerful feature learning and feature expression abilities than traditional machine learning methods, which is useful in improving classification accuracy by capturing complex and abstract features of images.[8], [9] CNNs have also been reported to be better than the traditional approaches at addressing the issue of "noise" (presence of irrelevant information in data) in image classification problems.[8] Different types of noise includes impulse noise (occurs during image acquisition or transmission), missing image samples (loss or occlusion of the desired parts of the image), packet loss (poor image quality), etc. resulting in damaged or lower quality images, making classification difficult.[10] Thus, in this project, we aim to develop and apply a multi-layer CNN for the task of image classification in a dataset of 10 different brand logos.

II. IMPLEMENTATION

Our implementation consisted of the following four steps:

Step I: Importing Data – After importing the data, we noticed some mislabeled data so, we performed dictionary mapping to correct the wrongly labeled images in the dataset by reassigning some old labels to new labels. Once the labels are corrected, the image dataset is reshaped into images of size 300x300 with 3 color channels (300x300x3) and this reshaped dataset is then split into a training and test set (80:20 split).

Step II: Building the Model and defining the Model Architecture – We developed our model using a robust pre-trained model as the foundation. While there are different pre-trained models[11] that could be used for logo classification problems such as VGG19,[12] ResNet50V2, etc.,[13], [14] the ResNet50V2 was chosen as it has a relatively complex architecture but still has a considerably small model size, making ResNet faster and computationally more efficient than its counterparts (VGG16, VGG19, etc.).[11], [13] The initial layers of our model consist of a Normalization() layer and a Resizing() layer to normalize the image inputs based on the mean and variance of each color channel based on CIFAR-10 dataset values and resize the images to match the expected input size of the ResNet50V2 model (224 x 224) respectively. All 50 layers of the ResNet50V2 were trained in our implementation. This pre-trained model is followed by a dropout regularization of 0.5 and BatchNormalization() layer to improve the stability of the model. We included seven additional layers to the CNN architecture following the pretrained model, to further improve the accuracy of our model in the validation/test dataset. The first additional layer, GlobalMaxPooling2D(), was added to reduce the spatial dimensions of the output. This was then followed by Dense layers with 1024 and 512 units, with BatchNormalization() and a Dropout layer (of 0.5 and 0.25 respectively) after each Dense layer. Our output layer was a Dense layer with 10 units using the SoftMax activation function which produces a probability distribution over the 10 classes.

Step III: Training the Model – We compiled the model using sparse categorical cross-entropy loss function and the Adam optimizer. The sparse categorical cross-entropy loss function was chosen as we are facing a multi-class problem and the target labels are integers. We trained the model using a batch size of 32 and 50 epochs. We used ModelCheckpoint() and EarlyStopping() callbacks from the Keras library to save the best model weights and stop training early if the validation loss did not improve after 10 epochs.

Step IV: Evaluation of Model Performance

In this final step, the performance of our model was evaluated on a test set using the evaluate() function of Keras. Overall, our implementation of the brand logo classification system using a CNN achieved an accuracy between 94 – 97 % on the test dataset and showed promising results in detecting brand logos in real-world images. To test the model on the “hard” test dataset, which includes images that belong to none of the 10 classes that the model was trained on, we used a probabilistic approach by setting a threshold value for the probability to 50% and assigned it to label “-1” for any prediction where the probabilities of the image belonging to any of the 10 classes was less than the threshold percentage; This approach allows us to maintain the simplicity and accuracy of the CNN developed for classifying the given 10 logos while also being able to also classify the “unknown” class.

III. EXPERIMENTS

Initially, we attempted to develop our own architecture for the multi-layer CNN, without using pre-trained models and we were unable to achieve an accuracy in the test set beyond 20%, even after data augmentation, which we attribute to simplicity of the model being unable to effectively learn the features necessary for classification. To remediate this issue, our next experiments involved transfer learning. Using pre-trained models as the base for our CNN made it so our model would not need to learn how to capture the lower-level features and the computation effort could be directed towards the specific task at hand.

Three transfer learning models we experimented with were VGG16, ResNet50V2, and ResNet101, chosen specifically for their high performance and robustness in image-classification, particularly logo classification problems.[13], [15] After testing with just the pre-trained model, there was a substantial improvement in validation accuracy from less than 30% to ~84% with VGG16, ~90% with ResNet50V2, and ~65% accuracy with ResNet101. Based on these results, we decided to use the ResNet50V2 as the base model for our algorithm and worked on refining this further by tuning the hyperparameters such as adding more layers, including batch normalization, and using dropout regularization in combination with the base model to further refine the model. In our next experiments we used only the top 10 layers of the ResNet50V2 and performed data augmentation, a technique for improving model generalizability, where the size of our training dataset is increased by creating "new" data by randomly rotating, shifting, shearing, zooming, and flipping images to train the model. The goal was that with "more" data in training would improve model accuracy and generalizability. In addition to this, we also tried to use multiple dense layers (2 - 9) and varied the number of units in each layer between 256 – 4096. This experiment resulted in an overly complex architecture with minimal improvement in validation accuracy, so we reverted to a simpler model. In our next set of experiments, we decided to not perform data augmentation due to its use increasing training time significantly without any improvement in performance. We then experimented with using all 50 layers of the ResNet50V2 and reducing the number of dense layers between the input and output layers to 3 or less. This CNN architecture yielded the highest validation accuracy (~ 95%) among all our experiments, so we used this architecture for our final implementation. However, there were indications of potential overfitting with this architecture, so we tried to address this by incorporating Dropout layers with values as high as 0.5, which resulted in a 2% increase in accuracy, so we kept those. Finally, we experimented with using various optimizers (Stochastic Gradient Descent, Adam, Nadam, etc.) to determine if one outperformed the others in convergence to the best weights. None of these experiments yielded significant improvement in accuracy.

IV. CONCLUSIONS

Based on the results of all experiments, the final model had an accuracy between 93 – 98% in the test dataset. In this project, data augmentation did not cause a significant increase in model accuracy when using the same number of layers in the architecture, which is why data augmentation was not implemented in the final model for this project.

The final model architecture was a multi-layer CNN with ResNet50V2 as the base model, followed by a global average pooling layer and then two successive dense layers with 1024 and 512 units and finally the output layer with 10 units along with Batch normalization and dropout after each of these layers. These results indicate that our model is successful in the classification of 10 different brand logos but testing with larger datasets is warranted.

V. ACKNOWLEDGEMENTS

We thank Dr. Catia Silva for her guidance and insights throughout the semester. We also appreciate Andrew Jensen and Bradley Johnson for their time and support. We would like to acknowledge the University of Florida Research Computing for providing computational resources and support that have contributed to the results reported in this project. URL:http://www.rc.ufl.edu

VI. REFERENCES

[1] H. Jahankhani, T. Jayaraveendran, and W. Kapuku-Bwabw, "Improved Awareness on Fake Websites and Detecting Techniques," in *Global Security, Safety and Sustainability & e-Democracy*, C. K. Georgiadis, H. Jahankhani, E. Pimenidis, R. Bashroush, and A. Al-Nemrat, Eds., in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Berlin, Heidelberg: Springer,

2012, pp. 271–279. doi: 10.1007/978-3-642-33448-1_36.

[2] G. Edwards, "Cybercrime: Targeting your intellectual property," *Intellectual Property Forum: Journal of the Intellectual and Industrial Property Society of Australia and New Zealand*, no. 124, pp. 35–40, Jun. 2021, doi: 10.3316/agispt.20210803051032.

[3] I. Fehérvári and S. Appalaraju, "Scalable Logo Recognition Using Proxies," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2019, pp. 715–725. doi: 10.1109/WACV.2019.00081.

[4] P. N. Druzhkov and V. D. Kustikova, "A survey of deep learning methods and software tools for image classification and object detection," *Pattern Recognit. Image Anal.*, vol. 26, no. 1, pp. 9–15, Jan. 2016, doi: 10.1134/S1054661816010065.

[5] A. A. M. Al-Saffar, H. Tao, and M. A. Talab, "Review of deep convolution neural network in image classification," in *2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, Jakarta: IEEE, Oct. 2017, pp. 26–31. doi: 10.1109/ICRAMET.2017.8253139.

[6] K. L. Chiew, E. H. Chang, S. N. Sze, and W. K. Tiong, "Utilisation of website logo for phishing detection," *Computers & Security*, vol. 54, pp. 16–26, Oct. 2015, doi: 10.1016/j.cose.2015.07.006.

[7] J. Kim, B.-S. Kim, and S. Savarese, "Comparing Image Classification Methods: K-Nearest-Neighbor and Support-Vector-Machines," in *Proceedings of the 6th WSEAS International Conference on Computer Engineering and Applications, and Proceedings of the 2012 American Conference on Applied Mathematics*, in AMERICAN-MATH'12/CEA'12. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2012, pp. 133–138.

[8] P. Wang, E. Fan, and P. Wang, "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning," *Pattern Recognition Letters*, vol. 141, pp. 61–67, Jan. 2021, doi: 10.1016/j.patrec.2020.07.042.

[9] S. Kumar, Z. Khan, and A. Jain, "A Review of Content Based Image Classification using Machine Learning Approach," *International Journal of Advanced Computer Research*, vol. 2, no. 3, pp. 55–60, Sep. 2012, [Online]. Available: https://login.lp.hscl.ufl.edu/login?url=https://www.proquest.com/scholarly-journals/review-content-based-image-classification-using/docview/1198000536/se-2

[10] M. Momeny, A. M. Latif, M. Agha Sarram, R. Sheikhpour, and Y. D. Zhang, "A noise robust convolutional neural network for image classification," *Results in Engineering*, vol. 10, p. 100225, Jun. 2021, doi: 10.1016/j.rineng.2021.100225.

[11] A. Kumar *et al.*, "Multilabel classification of remote sensed satellite imagery," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 7, p. e3988, 2021, doi: 10.1002/ett.3988.

[12] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition." arXiv, Apr. 10, 2015. doi: 10.48550/arXiv.1409.1556.

[13] S. Sahel, M. Alsahafi, and T. Alsubait, "Logo Detection Using Deep Learning with Pretrained CNN Models," *Engineering, Technology & Applied Science Research*, vol. 11, pp. 6724–6729, Feb. 2021, doi: 10.48084/etasr.3919.

[14] D. Sarwinda, R. H. Paradisa, A. Bustamam, and P. Anggia, "Deep Learning in Image Classification using Residual Network (ResNet) Variants for Detection of Colorectal Cancer," *Procedia Computer Science*, vol. 179, pp. 423–431, Jan. 2021, doi: 10.1016/j.procs.2021.01.025.

[15] M. Bansal, M. Kumar, M. Sachdeva, and A. Mittal, "Transfer learning for image classification using VGG19: Caltech-101 image data set," *J Ambient Intell Human Comput*, vol. 14, no. 4, pp. 3609–3620, Apr. 2023, doi: 10.1007/s12652-021-03488-z.