**CAPSTONE PROJECT INTERIM REPORT**

**Title: Fake Brand Logo Detection**

**Authors**

Delphina Dominic Carval (0787542)

Data Analytics for Business

St Clair College, Windsor, Ontario

Akshay Shaji (0797556)

Data Analytics for Business

St Clair College, Windsor, Ontario

Deelan Joston Monteiro (0803137)

Data Analytics for Business

St Clair College, Windsor, Ontario

Sai Kumar Yamasani (0795741)

Data Analytics for Business

St Clair College, Windsor, Ontario

**Introduction:**

Fake brand logos, also known as counterfeit or knockoff logos, are logos that are intentionally designed to look like the logos of well-known brands but are not authorized or produced by the brand owner. These logos are often used to sell fake or low-quality products to unsuspecting consumers, which can harm the reputation of the brand and the safety of the consumer.

The problem of fake brand logos is widespread and can affect a wide range of industries, from fashion and luxury goods to electronics and pharmaceuticals. In fact, the global value of counterfeit goods is estimated to be in the hundreds of billions of dollars annually.

**Importance:**

Detecting fake brand logos is important for several reasons.

* + Firstly, it helps to protect the brand owner's reputation and revenue by preventing counterfeit products from being sold under their name.
  + Secondly, it helps to protect consumers from being deceived into purchasing fake or low-quality products that may not meet safety standards.
  + Finally, it helps to support fair competitionbypreventing counterfeiters from undercutting legitimate businesses with cheaper, inferior products.

**Objective**:

The fake brand logo detection project aims to develop a system that can identify fake logos and distinguish them from the original product. The project will use machine learning and deep learning techniques to build a fake logo detector website and assess how much they resemble the original product logo. The project's goal is to help consumers verify whether a product is original and to help brands combat logo piracy. Counterfeit products can harm a brand's reputation and sales, and consumers can be cheated out of their money. The logo detection system will help brands protect their brand identity and prevent fraud by detecting and eliminating e-commerce listings containing fake logos. The project is beneficial for both consumers and brands.

**Related Research:**

There is research done by SRM Institute of Science and Technology in which they have used correlation method and Tesseract-OCR to detect fake logos. Tesseract-OCR is a tool that can recognize text in images. It works well, but sometimes has trouble with unique fonts due to which they have 80% accuracy. After thinking about doing a project on same topic and when we were thinking how we can increase the accuracy, we came across beer logo detection project done by Meerkat Cv, a computer vision tech company specialized in face recognition and object detection. In their project they have used machine learning. They have trained their system to detect logos of the following beer brands: Budweiser, Bud Light, Corona, Guinness, Heineken and Stella Artois on social media. After going through their project we thought to use machine learning in detecting fake brand logos keeping in mind that machine learning algorithm works well with unique fonts as compared to Tesseract-OCR. We will be using machine learning algorithms to increase accuracy by more than 80%.

**Dataset:**

Our dataset consists of 6000 images of 50 brands with 100 original images and 20 fake images of each brand. We analyzed the logos2k+ and Flicks dataset. In these two datasets there are images of only original logos. Whereas for our project we needed both fake and original images. We didn’t have any proper dataset according to our requirements. So, we created our own dataset. The images are collected from various sources such as the original logos we downloaded the images from every brand’s official site and from google. While in case of fake logos, some brands fake logos were easily available on google and Pinterest, so we downloaded from these sites and for brands whose fake logos were not available, we generated the logos. We generated the fake logos using “Brand crowd” website and PowerPoint. All the downloaded images were in different format for e.g. Some images were in JPEG while some were in PNG format. So, by using PIP library from python, we converted format of every image to JPG. After this we renamed all the images as some pattern. We have created two separate folders “fake” and “original” under the “brand's” folder in google drive and uploaded all the images. We have used the collaboration Python notebook for coding and imported all the images from google drive.

**Five c’s**

**Consent-** We ensure proper consent throughout the progress of our project. We took several steps to ensure consent for our project. We checked the terms and conditions of the websites to use the images when we were collecting images from the websites. We clearly explained information about our project, i.e., the purpose of our project and how we plan to use the dataset in our objectives. And ensure that the images used in the project are kept protected and confidential i.e. we will use the images only for our project purpose and not for any other unintended purpose.

**Clarity**- The purpose of the project is to detect fake brand logos. The website that we are developing will be designed only to identify fake logos and prevent their use and not be used to infringe on the rights of legitimate brands. The criteria for identifying fake logos will be based on the similarity of the logo to the original.

**Consequences**- Consequences is an important part of detecting fake brand logos, it makes companies and consumers protected from fraud and any misleading data. Our dataset is private and there is no chance that our dataset is misused. It also makes the companies make sure that the data is safe.

**Control**- We ensure that the project is carried out in a responsible and ethical manner. We ensure the dataset is used only for its intended purpose and is not misused for any other purpose. we will make sure that only our group members and professor have access to the dataset as it has fake logos of brand which our generated by us and misusing them can harm brands reputation. We will have a data-sharing agreement when we share files with third parties.

**Consistency**- Consistency is an important aspect of fake brand logo detection. Our dataset contains fake and original images of different brands, where each image with the same format and unique name. It’s very easy to identify each brand in the dataset. By using this model can be trained easily and processed the results are accurate and reliable.

**ETL**

We have created ‘target\_dict’ to map labels of the dataset ("Original" and "Fake") to integer labels (0 and 1). After that, we created two empty lists for storing images and corresponding for its labels. Then we create a ‘for’ loop to iterate over all the subdirectories in the "Brands" directory and process the images in each subdirectory based on their corresponding label. It allows the code to prepare the image data for the model by organizing the images into separate classes based on their labels. The ‘current\_label’ variable is later used to assign integer label to each image, which is necessary for training the model to recognize the different classes of images. We have created ‘for’ loop and ‘current\_label’for the subfolder also. We put ‘if img.is\_file ()’ for checking the file.). We loaded all the images as grayscale images using load\_img(img, color\_mode = "grayscale"). After that, We converted all the images into a Numpy array using ‘img\_to\_array(img)’ and using smart\_resize() all the images are resized to 256x256 pixel size. The code‘images.append(img\_array\_resized)’ add the resized image array to the list of images and ‘labels.append(target\_dict[current\_label])’ adds the corresponding label for the current image to the list of labels using ‘ target\_dict ‘ dictionary to map the label to an integer label. We calculated number of images using the ‘len()’ function and we have used train\_test\_split() from scikit-learn library to split the images and labels into training and testing sets(20% of the data is used for testing and 80% for training).We also converted values to the numpy array and scale values in [0,1] interval. After that, we flatten the image data to prepare it for use as features in a model.

**MODEL DEVELOPMENT**

**SVM MODEL**

the SVM model is used to classify data into two categories - 'fake' and 'original’. The model is trained to classify images into two classes - 'fake' and 'original'. The SVM model is trained using the 'linear' kernel, the regularization parameter 'C' is set to 1, which controls the trade-off between maximizing the margin and minimizing the classification error. The random state parameter is set to 42.

After training, it is evaluated on a validation set and a test set using the predict method to measure its performance in classifying new data. The validation set and test set both contain data with known labels of 'fake' or 'original', which are used to assess how well the SVM model can classify new data into these categories.

The accuracy score is calculated for both the validation set and the test set.

**Validation accuracy: 0.8175**

**Test accuracy: 0.845**

After this we have calculated precision, recall, and F1 score for the predicted labels for Validation and test datasets.

**Validation precision: 0.8045191900806326**

**Validation recall: 0.8175**

**Validation F1 score: 0.8100635927302098**

The validation precision of 0.8045191900806326 and recall of 0.8175. means that the precision and recall are around 0.80-0.82%, which indicates that the model is able to identify positive cases (i.e., fake images) with a high degree of accuracy. The F1 score is around 0.81, which suggests that the model is able to balance precision and recall reasonably well.

**Test precision: 0.8380048076923078**

**Test recall: 0.845**

**Test F1 score: 0.8411936036550542**

the model achieved a test precision of 0.84, which means that 84% of the predictions made by the model for the positive class were correct. The test recall is 0.85, which means that 85% of the actual positive examples in the test set were correctly identified by the model. F1 score is 0.84, which indicates that the model has a good balance of precision and recall and is performing well overall.

The confusion matrix of the model's predictions on the test set

array ([[937, 83], [103, 77]])

Chart, waterfall chart

Description automatically generated

In this case, there were 937 true negative predictions, 83 false positive predictions, 103 false negative predictions and 77 true positive predictions.

**Precision recall curve**

**Line chart

Description automatically generated with low confidence**

The above plot shows that when threshold is set to 0.1, the precision slightly increases and recall slightly decreases.

Chart, bar chart, waterfall chart

Description automatically generated

The precision, recall and F1 score for original class is higher as compared to Fake class.

**CNN MODEL**

We have developed a CNN model using the Keras API in TensorFlow to classify images of brand logos as original or fake. The model architecture consisted of 6 convolutional ,2 max-pooling layers, 1 flatten layer, 2 dense layers along with 2 dropout and 2 batch normalization layers to prevent overfitting. The dataset was preprocessed to ensure that the images are of consistent size. The training dataset consisted of images of both authentic and fake brand logos, while the validation and test datasets were used to evaluate the model's performance. The model is initialized using the “HeUniform initializer”, with the input shape (256, 256, 1). The activation function used for all the convolutional layers is ReLU. The padding used is "same", which means the input and output dimensions are the same. The first two convolutional layers have 128 filters of size (3, 3). Then a max-pooling layer is applied to downsample the feature maps by a factor of 2. A dropout layer is added to avoid overfitting, and batch normalization is performed to normalize the activations between layers.

The next two convolutional layers have 128 filters of size (3, 3), followed by another max-pooling layer and a dropout layer with a rate of 0.2. Then batch normalization is performed again. The output is then flattened and fed into a fully connected dense layer with 128 units and a ReLU activation function. Batch normalization is performed before the final dense layer, which has two units and a SoftMax activation function. The model is trained for 20 epochs with a batch size of 25, using the RMSprop optimizer with categorical cross-entropy loss function. The model's accuracy is used as the primary metric to evaluate its performance on the validation and test datasets. The results showed that the model was able to accurately classify images of brand logos into their respective categories.

**Test accuracy : 0.8675000071525574**

**Validation accuracy: 0.8583333492279053**

We have created two plots to visualize the performance of the model during training. The first plot shows the training and validation accuracy per epoch. The second plot shows the training and validation loss per epoch.

Table

Description automatically generated

Table

Description automatically generated with medium confidence

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

**Precision, Recall and F1 score for Fake and Original**

precision recall f1-score support

0 0.91 0.94 0.92 1020

1 0.57 0.47 0.52 180

accuracy 0.87 1200

macro avg 0.74 0.70 0.72 1200

weighted avg 0.86 0.87 0.86 1200

The "precision" for class 0 is 0.91, which means that out of all the samples predicted as class 0 by the model, 91% of them were class 0. The "recall" for class 0 is 0.94, which means that out of all the actual class 0 samples, the model correctly identified 94% of them. The "f1-score" for class 0 is 0.92, which is the harmonic mean of the precision and recall. The "precision" for class 1 is 0.57, which means that out of all the samples predicted as class 1 by the model, 57% of them were actually class 1. The "recall" for class 1 is 0.47, which means that out of all the actual class 1 samples, the model correctly identified 47% of them. The "f1-score" for class 1 is 0.52The "accuracy" of the model on this dataset is 0.87, which means that the model correctly classified 87% of the samples. The "macro avg" and "weighted avg" are the average precision, recall, and f1-score across all classes, weighted by the number of samples in each class. The "macro avg" gives equal weight to each class, while the "weighted avg" gives higher weight to the class with more samples. In this case, the "macro avg" and "weighted avg" f1-scores are 0.72 and 0.86, respectively.

Confusion matrix

[[956 64],[ 95 85]]

Chart, waterfall chart

Description automatically generated

In this case, there were 956 true negative predictions, 64 false positive predictions, 95 false negative predictions and 85 true positive predictions.

Chart, bar chart

Description automatically generated

The precision, recall and F1 score for original class is higher as compared to Fake class.

**Conclusion:**

plan for sem4:

In the 3rd semester, we developed CNN and SVM models for 50 brands. For the SVM model, we got a Validation accuracy of 82% and a Test accuracy of 84%. For the CNN model Test accuracy of 87% and Validation accuracy of 86%. In the 4th semester we will work on more 50 brands and try to get good accuracy. we will use data augmentation technique to increase the number of images and deploy the model using the website.

**Contribution:**

**Dataset**

Delphina, Akshay, Deelan, Sai

**ETL**

Delphina, Akshay, Deelan

**Model Development**

Delphina

**Report**

Delphina, Akshay, Deelan

**Reference:**

[https://www.researchgate.net/publication/354507140\_Logo\_Infringement\_Detection\_using\_Machine\_Learning](https://www.researchgate.net/publication/354507140_Logo_Infringement_Detection_using_Machine_Learning%20)

[https://www.researchgate.net/publication/312194349\_Deep\_Learning\_for\_Logo\_Recognition](https://www.researchgate.net/publication/312194349_Deep_Learning_for_Logo_Recognition%20)

[https://scikit-learn.org/stable/modules/svm.html#classification](https://scikit-learn.org/stable/modules/svm.html%23classification)

<https://www.tensorflow.org/tutorials/images/cnn>

<https://keras.io/api/layers/convolution_layers/convolution2d/>

<https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html>

**Appendices:**

<https://github.com/Delphina08/Fake-brand-logo-detection>