Shopee price match guarantee

*CMPE 256 Web and big data mining, Spring, 2021
Instructor: Dr Magdalini Eirinaki,
Computer Engineering Department
San Jose State University, San Jose, CA

Akash Aggarwal
SID: 013856531
akash.aggarwal@sjsu.edu
Computer Engineering Department
San Jose State University

Sowmya Mruthyunjaya
SID: 014619930
sowmya.mruthyunjaya@sjsu.edu
Computer Engineering Department
San Jose State University

Sudha Vijayakumar
SID: 014749488
sudha.vijayakumar@sjsu.edu
Software Engineering Department
San Jose State University

***Abstract*—Most shoppers don't like to overpay for the products they buy online. They shop around various retailers for the same product before making the final purchase to get the best deal. While e-businesses have made life easy for people where they can buy products from the comfort of their home, they are also entangled in a price war with their competitors. Every business wants to retain or add more customers and, in the due process try to provide the best possible pricing. When people try to sell products online, sometimes duplicates are posted with different titles and pricing which cuts into the profits of the business. Companies use methods like marketing, advertisements, giveaways, fun competitions to attract customers. One of the ways, a company can offer its products at competitive rates to its customers is by-product matching. With the increase in the number of sellers and vendors, product matching has become a difficult task. In this project, we are [1] leveraging machine learning models to perform product matching so as help find a better way of providing the best price to the customers.**

***Index Terms*—product matching, price match, machine learning, competitive rate**

I. Introduction

In recent times, we have seen applications that can predict similar products based on the thumbnails provided. For example, Amazon provides recommendations based on similar titles and images of a product. The two types of recommendations are user-based and item-based collaborative filtering. In user-based collaborative filtering, similar users are matched according to the items that they like or have previously liked. In item-based collaborative filtering, similar items are matched using the metadata of the users that have rated an image. In this project, we are using item-based collaborative filtering along with image processing.

This project primarily offers a 'Price Match Guarantee' solution to the Kaggle Challenge Shopee-Price Match Guarantee to determine if two products are the same by examining their images. Typically, we think the motive of customers when they do online shopping are to buy products from websites that provide the best price. Therefore, retail companies aim to offer the best price for a particular product. Shopee is one of the leading e-commerce giants in Taiwan and Southeast Asia. They aim to provide an easy, fast, and secure online shopping experience specific to the region. Shopee is a pioneer in providing rigid payment and logistic support. They also offer a Lower Price Guaranteed option for various products to maintain customer satisfaction and retention. This challenge aims to provide efficient product matching that can support:

- More accurate product categorization.
- Relevant product recommendations.
- Uncover marketplace spam.

Also, customers will be able to make an informed decision in choosing the right product with the best price guarantee. To achieve this, we plan to perform product matching using hybrid similarity scoring techniques coupled with deep learning algorithms with the given product images and metadata. In this project, we are using image-based deep learning techniques like Resnet50, Restnet152, [2]EffnetB3, EffnetB5, EffnetB6, Eca-nFnet-10 along with Tf-Idf Vectorization. We have done Match-by-phashing to find the baseline score. Finally, [3]ensemble learning has been performed to find the best set of working models.2

II. Approach

Fig. 1. gives a high-level overview of our project. The flow of the project is as follows:

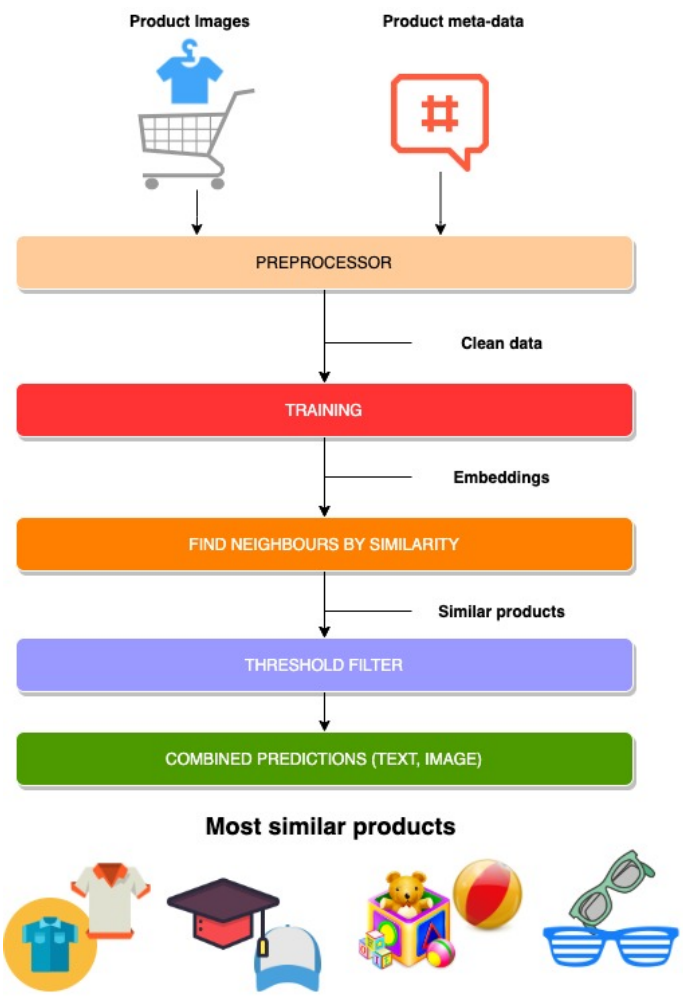


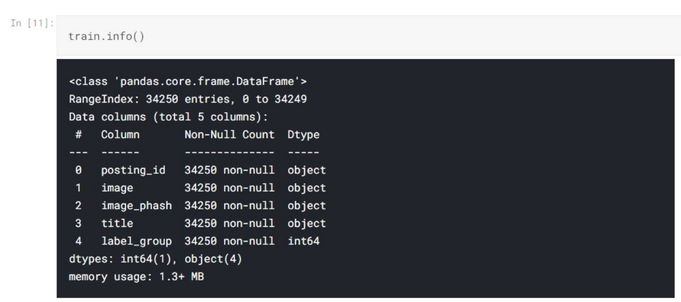
Fig. 1. High-level project pipeline

- Exploratory Data Analysis and preprocessing - This section will discuss the basic EDA performed on the training data set and extract more useful information about the attributes provided in the training data.
- Model selection - [4]This section describes the various models we have used in our project. It also shows the hyperparameters that are tuned to beat the baseline and the performance of the chosen models.
- Experimental study -In this section, we will discuss the steps of implementations required in our project.
- Result analysis - This section will analyze the results from our implementation of the deep learning models. We will evaluate the performance based on the F1 score of all the models.
- Conclusion - This section concludes the model that worked best for the project, results obtained. It also includes a summary of the future works to improve the performance of our work.

III. Exploratory data analysis

In any machine learning problem, it is always good to first understand the data and get as many insights as possible before preprocessing it or feeding it to any model for training. Since our team entered the kaggle challenge, the dataset was available as part of it. We performed our initial investigation on the data to see if we can uncover any anomalies, find any correlation between features, generate summary statistics, etc using the PANDAS Data Profiling tool.

The dataset contained 2 components: 1) Product meta-data in CSV format and, 2) Product images. The overall size of the dataset is 1.79 GB. The following figures and tables outline some detailed information about the dataset.



```
In [11]: train.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34250 entries, 0 to 34249
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   posting_id   34250 non-null  object
 1   image        34250 non-null  object
 2   image_phash  34250 non-null  object
 3   title        34250 non-null  object
 4   label_group  34250 non-null  int64
dtypes: int64(1), object(4)
memory usage: 1.3+ MB
```

Fig. 2. Summary of train meta-data

Dataset	Size
Training	34250
Test	3

TABLE I
SIZE OF THE SHOPEE DATASET

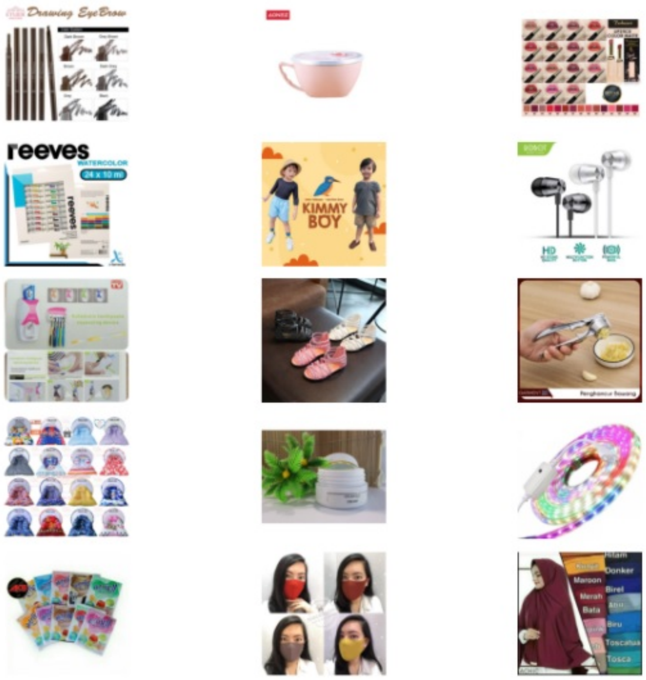


Fig. 3. Sample of train images

Fig. 4. Sample of test images

Fig. 3. and Fig. 4. show sample training and testing images. Fig. 5. provides insight into the metadata of the project. It shows the top 50 duplicate items grouped by label-group or product category present in the training set.

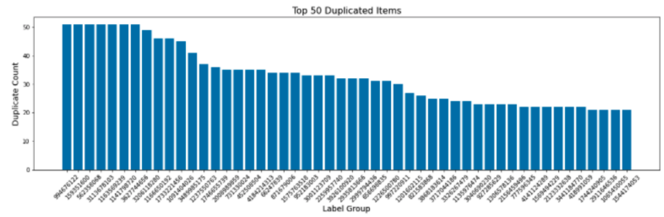


Fig. 5. Top 50 Duplicated Items according to label

Fig. 6. provides the relationship between duplicate and unique item count. It shows that as the index of unique item increase, the duplicate count decreases.

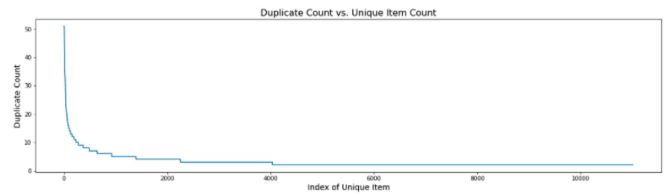


Fig. 6. Duplicate Count vs Unique Item

Fig. 7. provides examples for same product images in the dataset whereas Fig. 8. highlights how same product images may look different,

Fig. 7. Same product - Same image

Fig. 8. Same product - Slightly different image

After performing the EDA, we found that there are multiple ways to find similar products:

- We can use title and find similar products by title.
- We can use product images and find a similar product using images.
- We can use image hash and find a similar product using images.
- We can use image OCR Text matching and find a similar product using images.
- We can merge image OCR Text along with the product titles and find a similar product by merged title.

IV. Feature Engineering Preprocessing

Fig. 9. outlines the [5]preprocessing steps carried out on the meta-data and images to extract a clean meta-data for further processing,

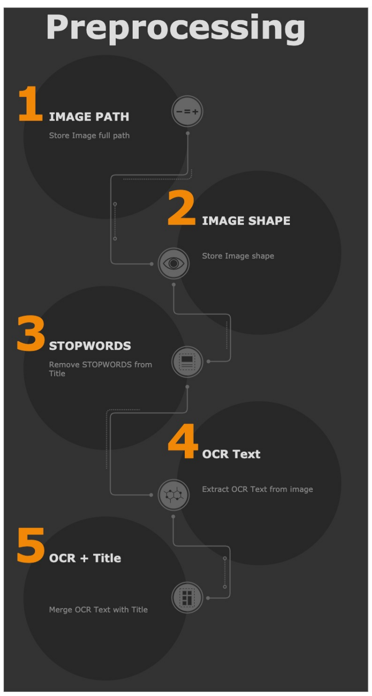


Fig. 9. Preprocessing pipeline

V. Model Selection

Since we are dealing primarily with images, we decided to choose some [6]CNN backbones that have provided exemplary results on various well-known image datasets.

A. Resnet

Resnet is the short form for the residual network which serves as the backbone for many computer vision tasks. Deep neural networks suffer from overfitting and vanishing gradient problems. Resnet addresses those problems through skip connections where the gradients from one layer are passed to another layer deep in the network through identity blocks.

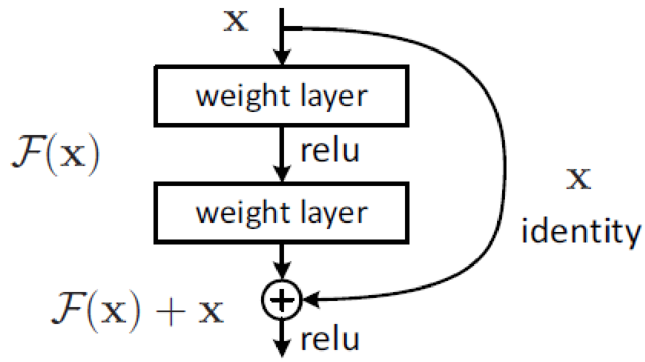


Fig. 10. Resnet

In this project we have used [7]resnet152 and [8]resnet50. The numbers 152 and 50 correspond to the number of layers in the network.

B. Efficient net

EfficientNet is a convolutional neural network that scales the depth (which is the number of layers in the network), width (number of filters in a convolutional layer), and resolution (height and width of the input image) of the network uniformly using a fixed compound scaling coefficient. The way compound scaling works is if the input image is bigger, more layers are needed in the network to increase the receptive field and more channels to capture more fine-grained patterns on the bigger image. Efficient provides very high accuracy while also improving efficiency by reducing parameters and

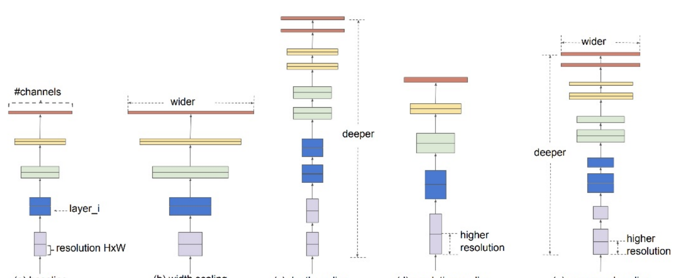


Fig. 11. Resnet

C. Efficient Channel Attention net (eca-nfet-10)

[9]Eca-nfnet is a pretrained model on ImageNet and eca-nfnet-10 is a variant of the NFNet (Normalization Free) model family. Eca net uses very few parameters and less computation to increase the network performance. It avoids dimensionality reduction and efficiently reduces cross-channel interaction. Eca-nfnet model variant results in an improved runtime characteristic, throughput on a GPU accelerator. It utilizes Efficient Channel Attention that is ECA instead of Squeeze-Excitation. It also features SiLU activations instead of the GELU.

Like other models in the NF family, this model contains no normalization layers (batch, group, etc). The models make use of Weight Standardized convolutions with additional scaling values in place of normalization layers.

VI. Experimental Study

Fig. 12. and 13. shows the flowchart of finding similar images by product title and product image. For finding similar images using product image, after the initial EDA and preprocessing the data, we applied transfer learning by using [10] [11]pre-trained models like resnet152, ecanfnet-10, efficient net [3,5,6]. We then generated image embeddings to bring down the dimensionality of these images. The key to good embedding is to train the model so that similar images are converted to similar vectors. K-nearest neighbor algorithm and a similarity measure like cosine similarity are then used to predict similar products. This list is then passed through a suppression filter to generate the most similar products. For finding similar images using [**?**]product title, the clean title is feed into the pre-trained models. The text embeddings from the previous step are compared by vector similarity to find similar products. The rest of the steps are similar to the Match-Product-ByImage pipeline.

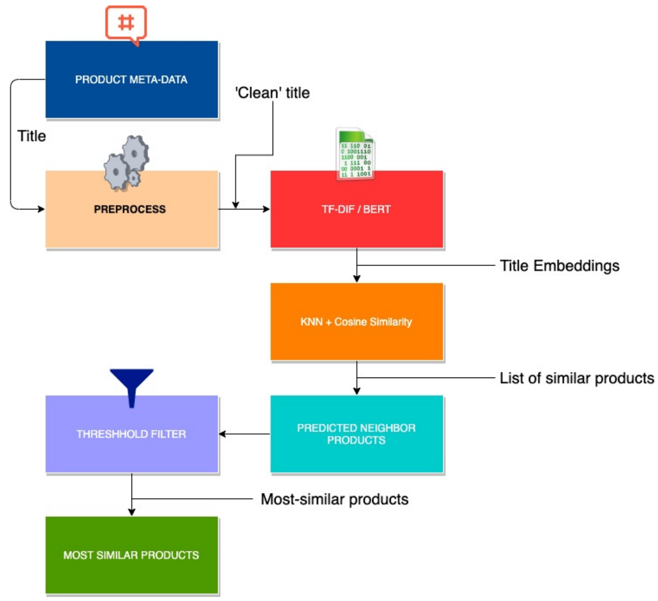


Fig. 12. Flowchart of Match-Product-ByImage

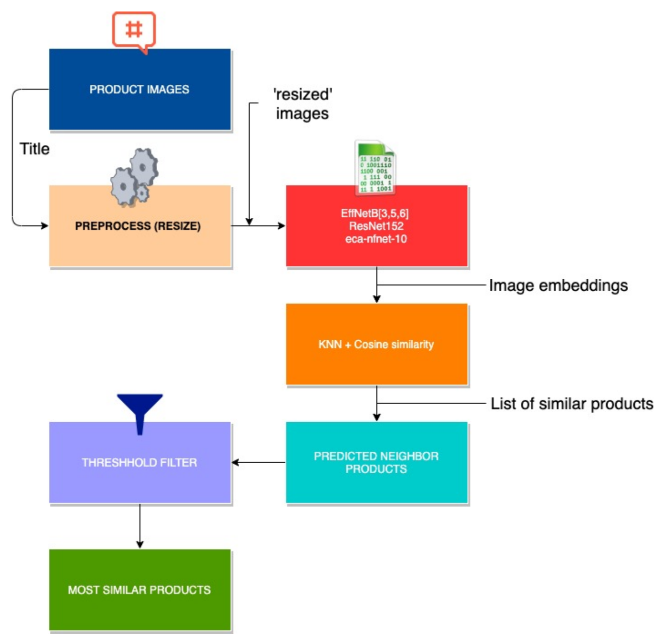


Fig. 13. Flowchart of Match-Product-ByTitle

Steps involved in project implementation:

- *Step 1:* Train Resnet152 and compute the F1 score.
- *Step 2:* To replace [RESNET152] model with the EfficientNet [3,5,6] models and train them on the SHOPEE Dataset, to see if the current [F1] Score can be improved.

- *Step 3:* To replace the [EfficientNet] model with the [eca-nfnet-10] model and make inferences on the SHOPEE Dataset, to see if the current [F1] Score can be improved.
- *Step 4:* Ensemble [EfficientNetB6] + [eca-nfnet-10] to see if the [F1]-Score improves.

VII. Result Analysis

In our project, we employed various deep learning models and found the following F1 score respectively: Table 1 shows the model vs F1 score:

Model Name	F1 Score
Simple PHash	0.585
ResNet152	0.69
EffnetB-3	0.712
EffnetB-5	0.723
Eca-nfnet-10	0.725
EffnetB-6	0.729
EffnetB-6 + Eca-nfnet-10	0.733

TABLE II
F1 Score for various models

The following figure shows a bar graph comparing the performance of all the models based on their F1 score.

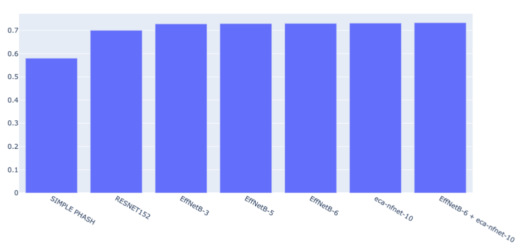


Fig. 14. Performance by F1 Score

The following figure shows a bar graph comparing the Time-to-train of all the models based on the number of epochs(20).

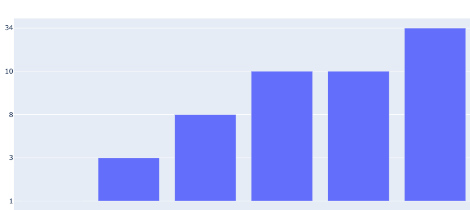


Fig. 15. Time to Train

VIII. Conclusion

After submission of our project to Kaggle(Team: Recommenders-4m-SPARTAN), we achieved the best test F1 score of 0.733, ranking us in the Top 11 percentile of all submissions in the competition so far. The model that performed the best was the ensemble model of EffecientB-6 and Eca-nfnet-10. Generally, ensemble models perform better than the standalone models and it was apparent in our result analysis as well. Image processing tasks are not only challenging but also time-consuming. While the results are satisfactory, given it is the first time we performed deep learning models for image processing, we intend to continue to improve the results. Especially, in terms of fast training and more accurate neighbor finding. We are trying to build the image processing backbone models with the help of fastai which considerably reduces the training time, almost 1/3 of the current high training time. And, to try the Spotify recommended method [12] ANNOY to find similar neighbors (products).

IX. Contribution

Individually all three of us spent some time looking for projects that we can work on, one that aligns with the course work of CMPE 256. After two rounds of brainstorming, we as a team decided to enter the Kaggle competition for Shopee Price Match Guarantee. After we got the nod from our course instructor, we officially entered the competition. We spent some time understanding the problem statement. Then we held regular project checkpoint meetings to address any concerns, share progress, exchange notes. As a team, we wrote the project proposal, explored the dataset, performed exploratory data preprocessing, drafted the project report, and prepared presentation slides. Individually we each tried to work on different models to see if we could beat the baseline of the challenge. Akash worked on Eca-nfnet-10, Sudha worked on Efficient-Net and Sowmya worked on Resnet-152 and finally, everybody worked on the ensemble modeling to achieve the best results.

X. Future work

- To try different ensemble with different neighbor thresh hold to see if the score improves.

- To try [Fastai + Annoy] (Match by image) along with BART tokenizer(match by title) to find similar products.
- To try domain-specific proven model Large Scale Multimodal Classification Using an Ensemble of Transformer Models and Co-Attention
- To implement product recommendation systems with the best-trained model as the prediction backbone.

XI. Acknowledgement

We would like to express our sincere gratitude to Dr. Magdalini Eirinaki for her guidance, knowledge, and patience throughout the course. She has answered all our queries with utmost patience and has guided us during times of need.

XII. Source Code

GitHub
How to RUN

XIII. ABBREVIATIONS

- EDA - Exploratory Data Analysis
- CNN - Convolutional Neural Network
- Resnet - Residual Networks
- EfficientNet - Efficient Networks
- eca-nfnet - Efficient Channel Attention net

References

[1] P. Ristoski, P. Petrovski, P. Mika, and H. Paulheim, “A machine learning approach for product matching and categorization,” *Semantic Web*, vol. 9, no. 5, p. 707–728, 2018.

[2] H. Alhichri, A. S. Alswayed, Y. Bazi, N. Ammour, and N. A. Alajlan, “Classification of remote sensing images using efficientnet-b3 cnn model with attention,” *IEEE Access*, vol. 9, pp. 14078–14094, 2021.

[3] V. Chordia and V. K. BG, “Large scale multimodal classification using an ensemble of transformer models and co-attention,” *arXiv preprint arXiv:2011.11735*, 2020.

[4] L. Ma, *Computationally efficient models for high-dimensional and large-scale classification problems*. PhD thesis.

[5] L. D. Nguyen, D. Lin, Z. Lin, and J. Cao, “Deep cnns for microscopic image classification by exploiting transfer learning and feature concatenation,” in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, IEEE, 2018.

[6] S. Bell and K. Bala, “Learning visual similarity for product design with convolutional neural networks,” *ACM Transactions on Graphics*, vol. 34, no. 4, p. 1–10, 2015.

[7] PyTorch, “Resnet-152.”

[8] S. Ray, “Disease classification within dermascopic images using features extracted by resnet50 and classification through deep forest,” *arXiv preprint arXiv:1807.05711*, 2018.

[9] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, “Supplementary material for “eca-net: Efficient channel attention for deep convolutional neural networks”,”

[10] Y. Yousfi, J. Butora, E. Khvedchenya, and J. Fridrich, “Imagenet pre-trained cnns for jpeg steganalysis,” in *Proceedings of the IEEE International Workshop on Information Forensics and Security, WIFS*, 2020.

[11] R. Wightman, “Pytorch image models.” https://github.com/rwightman/pytorch-image-models, 2019.

[12] J. Wan, S. Tang, Y. Zhang, J. Li, P. Wu, and S. C. Hoi, “Hdidx: High-dimensional indexing for efficient approximate nearest neighbor search,” *Neurocomputing*, vol. 237, pp. 401–404, 2017.