Combating Human-Trafficking: Using Scene Recognition to Identify Hotels

Abstract

Human trafficking is a form of modern-day slavery which affects up to 1.2 million children worldwide every year. Victims are often photographed in hotel rooms, hence identifying these hotels is important to support trafficking investigations. Given an image of a hotel room interior, we would like to identify the hotel from the TraffickCam dataset, which is a large gallery of training images with known hotel IDs. Identifying and classifying hotel room interiors can be considered a scene recognition/identification task, which is complex as images belonging to the same hotel may look very different and images belonging to different hotels may look very similar. We propose to use state of the art deep learning models for scene recognition to identify hotel rooms based on query images, and generate embeddings for these query images. These embeddings are then mapped to a hotelID using nearest neighbor match to the embeddings from the training set. For any test image, we should predict the most similar hotel ids sorted in the order of relevance.

1. Introduction

Human trafficking is estimated to have trapped 24.9 million victims.[4]. The pictures of these victims are often clicked in hotel rooms. Identifying these hotels is vital in the combat of human trafficking. We are trying to build a model that identifies the hotel in which the pictures of the victims had been clicked. This model can aid in human trafficking investigations to identify the locations in which the victims were pictured.

The problem was presented at the FGVC8 workshop [13]. This is more complex than a simple image recognition task, as there are a huge number of classification labels which is the number of hotels in the dataset. Also rooms within the same hotel may not look the same and rooms in different hotels (part of the same chain) may look very similar. Hence we may have high intraclass variance and low interclass variance. It becomes imperative to minimize the search space efficiently in order to identify possible matches within a reasonable margin of error. We have used a ResNet50 [5] based model to generate embeddings for the training and test dataset.

2. Related Work

A baseline approach for this challenge is already laid out by the paper that presented this challenge [6]. The model here is a ResNet50 model that generates embeddings for training and test images. These test embeddings are then mapped to hotelIDs using the nearest training embeddings.

The research team at Nexocode [3] used transfer learning with fastai to classify hotel images. Top layers of ResNet50 (trained on ImageNet data) were used for transfer learning, while fastai was used to normalise data. Using dropout, weight decay regularisation, and data augmentation, they were able to achieve a validation accuracy of 70.4%, rising to 80% with further data cleaning.

Ma et al [8] showed that CNNs can identify more complicated features in hotel images such as rectangles or circles in intermediate layers, which can be stacked for more complex scene recognition. A 152 layer ResNet model was used for final image feature representation.

Nvidia [12] experimented with a triple hinge loss to compare images by learning and understanding aesthetics. Hotels from the same chain should follow similar aesthetics and hence this study becomes relevant for identifying hotels.

Scene recognition is an image recognition task whose aim is to classify the location of the place where image is taken. Scene recognition is challenging as there may be ambiguity between classes - scenes may share objects, which leads to misclassification [7]. Current state of the art approaches utilise CNNs for scene recognition, although performance is still far less than SOTA approaches other recognition tasks (e.g., object or image recognition). FOSNet framework based on ResNet backbone is current state-of-the-art for scene recognition [11]. It is observed that since the task of scene recognition is not entirely subjective due to the nature of the scene images and the scene categories overlap, no one particular method can be generalized to all scene recognition tasks [10].

Hotels-50k [14] is a dataset of over 1 million images of hotel rooms from over 50,000 hotels worldwide. This dataset can be used to perform data augmentation which will increase the amount of training data.

3. Approach

3.1. Dataset

The primary training dataset we will be using is the 2021 Hotel-ID to Combat Human Trafficking Competition Dataset [6]. This dataset consists of photos of hotel room interiors without any people present. The training set consists of a total of 97554 images from 7770 hotels. The training data is divided into sub folders with a dedicated sub folder for each of the 92 hotel chains that the hotels are part of. We will be splitting this data into training and validation sets and use validation set for evaluating model performance.

Additional model training can be done using the Hotels-50k dataset [14], which contains around a million annotated hotel images and was also built for combating human trafficking.

3.2. Data Preprocessing

Following data preprocessing was done on the training data -

- Input images resized to 256x256 pixels from 1024. This reduced the size of the training data from initial 26 GB to around 8GB.
- *hotel_id* column is label encoded.
- Data augmentations like flipping, rotation, brightness, etc. was be applied on the data.
- Data was processed in batches of 128.

3.3. Model

The model uses a ResNet50 [5] architecture that is pretrained on ImageNet [1] dataset. We trained this model to generate embeddings for input images. Embeddings are generated for the training images as well. We then find the 5 most similar embeddings from the training set and return the hotel IDs of these 5 images in the training set. We also experimented with an EfficientNet backbone to compare performance.

3.4. Loss Function

According to our data analysis, the data has high intra-class variance and low inter-class variance. Additive Angular Margin Loss (ArcFace) [2] is a function proposed for face recognition tasks, which can fit in this scenario as well. ArcFace loss is constructed by modifying the softmax loss.

The distance between two images can be calculated using cosine distance θ, which is the inner product of two normalized vectors. The features and layer weights are normalized using L2 normalization and inner product is calculated, giving us $\cos\theta$. Mathematically, ArcFace transforms logits $W_j^T x_i = ||W_j||\,||x_i||\cos\theta_j$, where θ_j is the angle between the weight W_j and the feature x_i. Additionally x_i is re-scaled to s after L2 normalization. An angular margin penalty of m is added to prevent weight of fully connected layer to be overly dependent on input data.

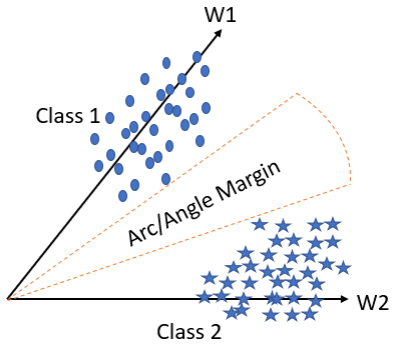

Figure 1: Geometric Representation of ArcFace

Final loss is given by applying softmax to $\cos\theta$ along with the added angular margin penalty.

$$L_3 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)}}{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}}$$

We used ArcFace [2] as the loss function to train our model. Pseudocode for ArcMax is shown below -

Algorithm : ArcFace Pseudocode

Input: Feature Scale s, Margin Parameter m, Class Number n, Ground-Truth ID gt

1. x = mx.symbol.L2Normalization (x, mode = 'instance')
2. W = mx.symbol.L2Normalization (W, mode = 'instance')
3. fc7 = mx.sym.FullyConnected (data = x, weight = W, no bias = True, num hidden = n)
4. original target logit = mx.sym.pick (fc7, gt, axis = 1)
5. theta = mx.sym.arccos (original target logit)
6. marginal target logit = mx.sym.cos (theta + m)
7. one hot = mx.sym.one hot (gt, depth = n, on value = 1.0, off value = 0.0)
8. fc7 = fc7 + mx.sym.broadcast mul (one hot, mx.sym.expand dims (marginal target logit - original target logit, 1))
9. fc7 = fc7 * s

Output: Class-wise affinity score $fc7$

We chose ArcFace for our model over other losses such as Triplet Loss or CosFace loss because it has empirically better inter-class discrepancy than Triplet Loss, while having about the same intra-class similarity. It also outperforms previous models using the other losses mentioned [9].

4. Experiments

4.1. Exploratory data analysis

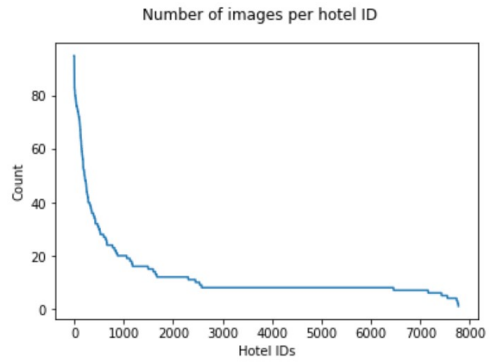

Figure 2: Number of images per Hotel ID

The number of images per class is not fixed. As seen in figure 2 the number of training examples is higher for hotel IDs 0 - 1000 and the number of examples significantly decrease after that.

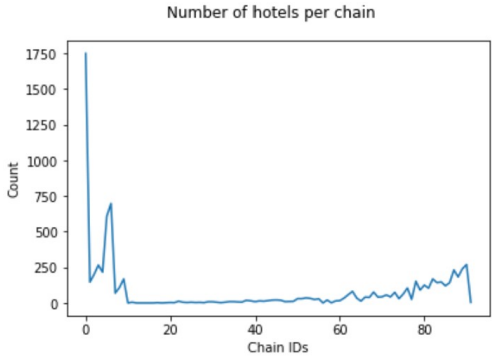

Figure 3: Number of hotels in a chain

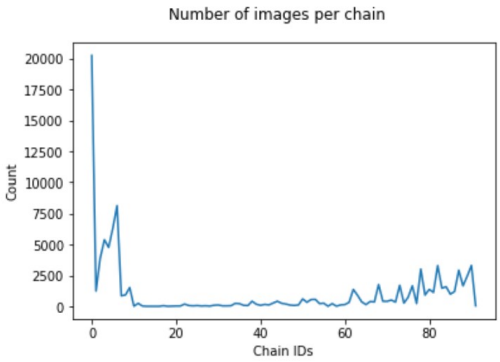

Figure 4: Number of images in a chain

The hotels are divided into chains and there are a total of 92 chains. As seen in figures 3 and 4 neither the number of hotels in per chain and nor the number of training examples per chain are fixed.

Rooms within the same hotel may not look the same as seen in figure 5 and rooms in different hotels (part of the same chain) may look very similar as seen in figure 6. Hence we may have high intraclass variance and low interclass variance. It becomes imperative to minimize the search space efficiently in order to identify possible matches within a reasonable margin of error.

Figure 5: Two images that belong to same class

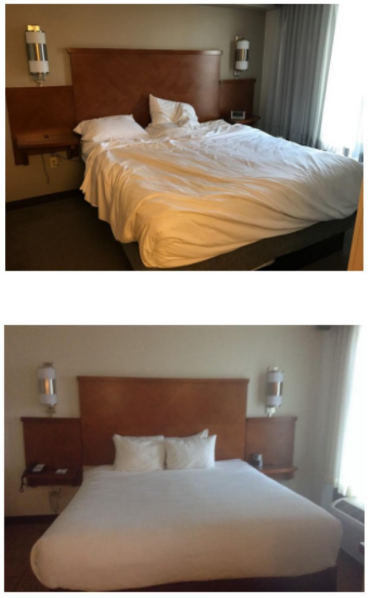

Figure 6: Two images that belong to different classes

Input image size	Hotels (Classes)	Epochs	LR	Model	Size of Embeds	Optimizer	MAP@5
256	7770	220	15e-5	ResNet50	256	SGD	0.3531
512	7770	6	1e-3	efficientnet_b1	1024	Adam	0.7151

Table 1: Results

4.2. Transformations

Following data transforms were performed on data before training -

- Horizontal and vertical flip.
- Shift, scale, and rotate.
- Optical Distortion.
- Perspective etc.

4.3. Model training

The input images were divided into training set and validation set in a 9:1 ratio. The input images were resized to 256x256 pixels. These images are then randomly cropped to get a 224x224 image. We use ResNet50 as the backbone for the model. With ArcFace [2] as the loss the model was trained using the below hyperparameters:

- Learning rate - 1.5×10^{-4}
- Batch size - 128
- Embedding size - 256
- Epochs - 220

The optimizer used was SGD with momentum. The loss decrease vs epoch is as shown in the figure 7.

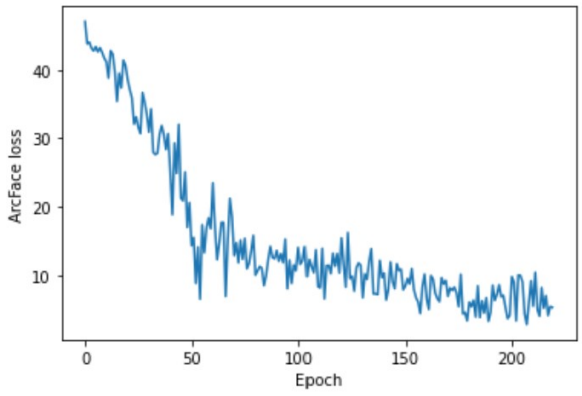

Figure 7: Loss vs epochs for the ResNet50 model

Similarly, we trained a model using EfficientNet backbone and following hyperparameters:

- Learning rate - 1×10^{-3}
- Batch size - 128
- Embedding size - 1024
- Epochs - 6

4.4. Evaluation

This is a classification problem. For each image in the validation set, we will predict a list of 5 hotel IDs that the image could belong to. This list is sorted so that the probability of the image belonging to the first hotel ID in the list is the highest.

Mean average precision @ 5 is the metric used to evaluate model performance. This is the same evaluation metric used by the competition [13]. The average precision @ 5 for an input image is calculated as below -

$$\text{AP@5} = \sum_{k=1}^{5} (Pr(k) \times id(k))$$

Here $Pr(k)$ is the precision at k and $id(k)$ is an identity function that evaluates to 1 if the k^{th} prediction is same as the true label and 0 otherwise. We find the mean of the AP@5 for all the input images to get the Mean average precision @ 5 for the model.

4.5. Results

For each image in the validation set, we used the model to generate embeddings. This embedding is then compared with the embeddings from the training set. 5 hotel IDs are then predicted by using nearest neighbor match.

The mean average precision @ 5 (MAP5) for the validation set was 0.35 for the ResNet model. Given the huge number of output classes (7770) these results are encouraging. Figure 8 shows 3 random images from the validation set and the predicted labels for those images. The images that are closest match to the query image is also displayed. This helped us analyze the model and understand the nuances associated.

The MAP5 accuracy for the model using EfficientNet backbone on validation set was 0.7151, which was considerably better. We observed the following differences led to improvements in performance (refer table 1 for hyperparameter comparison between both models) -

- Additional data augmentations

Figure 8: Five predictions for 3 randomly selected images in the validation set. The query images are in the first column. The 5 images besides the query image are images with embeddings closest to that of the query image. The true labels of all the images are above the image.

- Adam optimizer (vs SGD)
- Bigger embedding layer (1024 $>$ 256)
- Bigger input image size (512 $>$ 256)

Considering that our model with EfficientNet backbone has better accuracy, we use it for testing. We get accuracy of 0.592 on test set.

5. Conclusion

In this project we tried to solve the problem of hotel identification using indoor scene recognition. We first performed input data pre-processing as described in the earlier sections and reduced the size of the input dataset from 26GB to 8GB. We then trained a ResNet50 and an EfficientNet model to generate embeddings for hotel images. Finally embeddings were generated using the model for both train and test set, and we mapped test embeddings to hotel IDs using nearest neighbor match. We found the EfficientNet model to be better performing and we were able to get accuracy of 59.2% on the test set.

5.1. Scope for Improvement

In this section we have outlined the few steps that could be performed to enhance the model performance. These were not used in our model, but can be checked to compare performance -

- Experiment with other backbones such as EfficientNetv2, VGG etc.
- Try different types of losses such as triplet loss, CosFace etc.
- Trying ensemble models for better performance.
- Use k-means clustering and instead of nearest neighbor match.
- Training data can be augmented using the Hotels-50k dataset [14]. Additional model training can be done using this dataset.

5.2. Next Steps

With a model ready for recognizing images, it can then be deployed and productionalized for actual use for trafficking investigations. An app built with the model hosted on a

web server for inference based on submitted images could be a good application of the code.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li jia Li, Kai Li, and Li Fei-fei. Imagenet: A large-scale hierarchical image database. In *In CVPR*, 2009. 2

[2] Jiankang Deng, Jia Guo, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *CoRR*, abs/1801.07698, 2018. 2, 4

[3] KORNEL DYLSKI. Transfer learning in practice. image classification for hotel images with fast.ai library, Feb 2019. 1

[4] Human Rights First. Human trafficking by the numbers. https://www.humanrightsfirst.org/resource/human-trafficking-numbers, 2019. 1

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1, 2

[6] Rashmi Kamath, Gregory Rolwes, Samuel Black, and Abby Stylianou. The 2021 hotel-id to combat human trafficking competition dataset. 2021. 1, 2

[7] Alejandro López-Cifuentes, Marcos Escudero-Viñolo, Jesús Bescós, and Álvaro García-Martín. Semantic-aware scene recognition. *Pattern Recognition*, 102:107256, Jun 2020. 1

[8] Yufeng Ma, Zheng Xiang, Qianzhou Du, and Weiguo Fan. Effects of user-provided photos on hotel review helpfulness: An analytical approach with deep leaning. *International Journal of Hospitality Management*, 71:120–131, 04 2018. 1

[9] Charudatta Manwatkar. How to choose a loss function for face recognition, Jul 2021. 2

[10] Alina Matei, Andreea Glavan, and Estefania Talavera. Deep learning for scene recognition from visual data: a survey, 2020. 1

[11] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Fosnet: An end-to-end trainable deep neural network for scene recognition, 2019. 1

[12] Appu Shaji. Understanding aesthetics with deep learning, Feb 2016. 1

[13] Abby Stylianou, Rashmi Kamath, Richard Souvenir, and Robert Pless. Hotel-id 2021. https://sites.google.com/view/fgvc8/competitions/hotel-id2021, 2021. 1, 4

[14] Abby Stylianou, Hong Xuan, Maya Shende, Jonathan Brandt, Richard Souvenir, and Robert Pless. Hotels-50k: A global hotel recognition dataset. In *AAAI*, 2019. 1, 2, 5