Hotel Recognition to Combat Human Trafficking

Yuyu Lin, Peng Chen, Chi On Ho
Stanford University
Stanford, CA
{linyuyu,pengc,hochion}@stanford.edu

Abstract

We built a CNN-based hotel recognition system to combat human trafficking in this paper. Hotel recognition helps combating human trafficking by predicting the precise geographic locations of victims given hotel room images posted on the criminal networks. Recognizing hotels with hotel room images is challenging because hotel rooms look similar and images are often occluded by victims of human trafficking. We heavily applied data augmentation techniques, including random erasing and automatic augmentation, to overcome these issues, and to mitigate the limited amount of data available to us. We also formulated hotel recognition as a deep metric learning task besides as an image classification task and trained the CNN with triplet loss, motivated by face recognition. Combining both formulations by multi-task training, the model achieved the best performance. We did experiments on the Hotel-ID 2022 dataset to evaluate our system both qualitatively and quantitatively, which showed that it successfully recognized hotels through hotel room images.

1. Introduction

Human trafficking is one of the most concerning global issues that could potentially be addressed by modern computer vision techniques. As there are an increasing number of images of victims available online, one prevailing approach is to predict the precise geographic location of the victim when a hotel room image is posted on an advertisement or among the criminal networks. Since hotel rooms are often places where victims temporarily stay, hotel recognition is therefore an important and time-sensitive task to combat human trafficking. Hotel recognition, however, is not as easy as it seems, as images of hotel rooms are often of low quality and were taken at uncommon camera angles. Moreover, since these indoor venues mostly share similar amenities, it requires a fine-grained image classification model to properly handle a large number of classes and potentially high intraclass and low interclass variation. Kaggle has held competitions for two years tasked with identifying the hotel seen in test images and the latest competition is of 2022[1]. What makes the task of recognizing the hotel even more challenging is that images of hotel rooms are often occluded by humans or objects. Therefore the 2022 challenge opted to use a set of masked images as the test set so that more refined models can be trained to remain viable in a real world scenario.

The 2021 version of this competition [10] summarized two approaches to the task: (1) a classification network trained with cross-entropy loss, and (2) deep metric learning. Deep metric learning means first training a network to measure similarities between images and then classifying images by nearest neighbors. The paper suggested that the first approach performed better than the second approach. However, it also pointed out that the second approach was more applicable to real-world applications. In real-world settings, the system should be able to accept newly collected images after deployment to recognize new hotels. The first approach cannot handle new hotels unless the model is retrained on the new data, but the second approach can simply update the database of hotel room images to achieve this goal. In this project, we chose the first approach to get a better score as the set of hotels was fixed in our settings and dataset, but we also implemented the second approach to evaluate it. In addition, we proposed to combine both approaches by training our model with the second approach as an auxiliary task of the first approach to boost the model performance.

The input to our model is the variable-sized RGB images of hotels. The output is a list of hotel IDs ranked by the scores output by a classifier, which can indicate the order of rescue or location priority. We found data augmentation was crucial for the system, because the images in the test set are masked while the images in the training set are not masked. We mainly applied two sets of data augmentation methods, random erasing [23] and automatic augmentation [2, 3, 12].

We did experiments on the dataset provided by the Kag-

[1] https://www.kaggle.com/competitions/hotel-id-to-combat-human-trafficking-2022-fgvc9/overview

gle competition. Due to limited computational resources available to us, we did not use any other data except that the backbones are pretrained with the ImageNet dataset [4], while the top entries on the leaderboard used external data like Hotels-50K [19] heavily. Therefore, we did not aim at out-performing them, but focused on improving the performance under limited computational resources, comparing various decision choices, and analyzing the strength and weakness of the models. Our analysis should also apply to models trained on larger datasets and provide insights for future works on hotel recognition.

2. Related Work

Indoor Scene Classification Indoor scene classification focuses on images of indoor scenes that contain multiple distinct objects, with different scales, sizes and laid across different spatial locations in a number of possible layouts. The indoor scene problem is complicated for larger variations in light, shape, layout and severer occlusions [11]. Various methods [7, 13, 16] have been actively explored. To accommodate the different scene types, both global and local spatial information should be holistically leveraged [16]. Hayat et al. [7] proposes a new learnable descriptor called "spatial layout and scale invariant convolutional activations" upon which a new convolutional neural network architecture is designed to incorporate a "spatially unstructured" layer to improve robustness against spatial layout deformations. Yabei et al. [11] proposed MAPNet, which is a Multi-modal attentive pooling network for RGB-D indoor scene classification. However, their methods deal with private family room images, which may not be transferable for hotel interior images, since home rooms are stylistically varied and personalized decorated, so the differences between the images are mainly the furniture in the scene. For the interior of the hotel, the furniture style is basically uniform.

Deep Metric Learning with Triplet Loss As there are low inter-class differences and there might be new classes unseen in the training set, intuitively, our task (Hotel-ID classification) is very much like face recognition. Therefore, we investigated deep metric learning with triplet loss method, which is widely used in person ReID [8]. Deep metric learning maps an image into a feature vector in a manifold space via deep neural networks [6]. Triplet loss was introduced by Weinberger and Saul [22], which helps when we want to learn distributed embedding by similarity and dissimilarity. Specifically, FaceNet [17] learns a mapping from face images to a compact Euclidean space using a deep convolutional network, such that distances in the embedding space directly correspond to face similarity. Instead of optimizing an intermediate bottleneck layer as in previous approaches, FaceNet uses a triplet-based loss for training. Daniel et al. [21] further deals with face images that are partially occluded and makes the convolutional neural network learn discriminative features from all the face regions more equally to achieve higher recognition rate in practice. They proposed a modified loss function called batch triplet loss that improves the performance of triplet loss by adding an extra term to the loss function to cause minimization of the standard deviation of both positive and negative scores. They proved that there is consistent improvement in the LFW benchmark than FaceNet. Inspired by their work, we transfer this method into hotel interior recognition problem and train our model with triplet loss.

Hotel Interior Classification to Combat Human Trafficking There is several research work utilizing AI to stop human trafficking, especially child and/or sex trafficking [9]. Hamidreza et al. [1] extend the existing Laplacian SVM to identify human trafficking activities from textual advertisements from the website "Backpage". Anti-trafficking nonprofit Seattle Against Slavery[2] used a conversational agent to pretend as a girl and talk with the buyer, in order to identify the traffickers.

The Workshop on Fine-Grained Visual Categorization (FGVC) has held two iterations of hotel recognition competitions in 2021 [10] and 2022. Last year's first place winner[3] got a score of 0.8622. The author used metric learning and nearest neighbor search. The image representations were trained by Arcface [5]. Three backbones (ResNeSt101e, RegNetY120, and Swin Transformer) were used to create models with various combinations of training and index sets. In the nearest neighbor search, only the TOP1 nearest neighbor similarity score for each hotel was used as confidence. Another top entry[4] also utilized Arcface. Their experiments proved our intuition about the similarity between hotel ID classification and face recognition, that it is a solid idea to use well-established face recognition methods to solve this problem. Furthermore, instead of using ResNet as the backbone, we used ResNet-50 as the baseline and EfficientNet-B5 as the backbone to balance the performance of the model and our budget.

3. Methods

3.1. Problem Statement

Technically, our objective is to recognize the hotels in which given images are taken. The hotels are identified by unique IDs $y \in \mathcal{Y}$ and the images are denoted as $I \in \mathcal{I}$. Our model learns a map from images to hotel IDs $f : \mathcal{I} \rightarrow \mathcal{Y}$, i.e., we are working on an image classification task. The dataset we used to train and test the model consists of pairs

[2]https://www.thelanternproject.org/
[3]https://github.com/smly/hotelid-2021-first-place-solution
[4]https://github.com/michal-nahlik/kaggle-hotel-id-2021

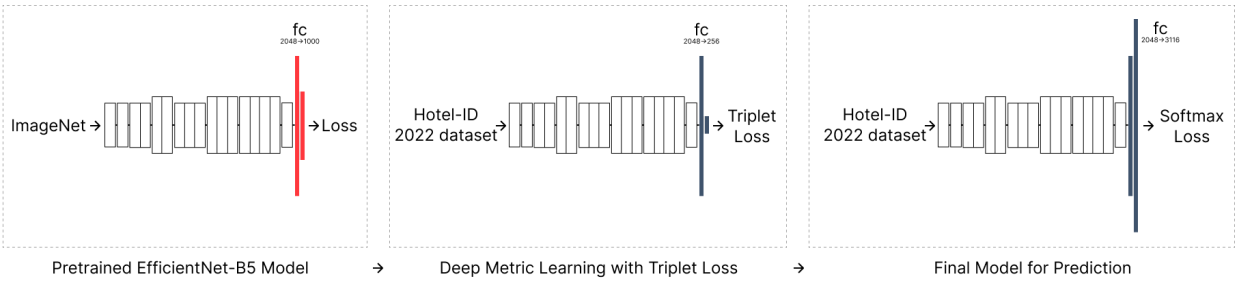


Figure 1. Model evolution. This is how we trained our model. Starting with the backbone EfficientNet-B5 pretrained on ImageNet, we replaced the last layer with a feature extraction linear layer and trained the whole network with triplet loss on the Hotel-ID 2022 dataset. Then, we replaced the last layer with a softmax classification layer and trained the network to classify hotel room images.

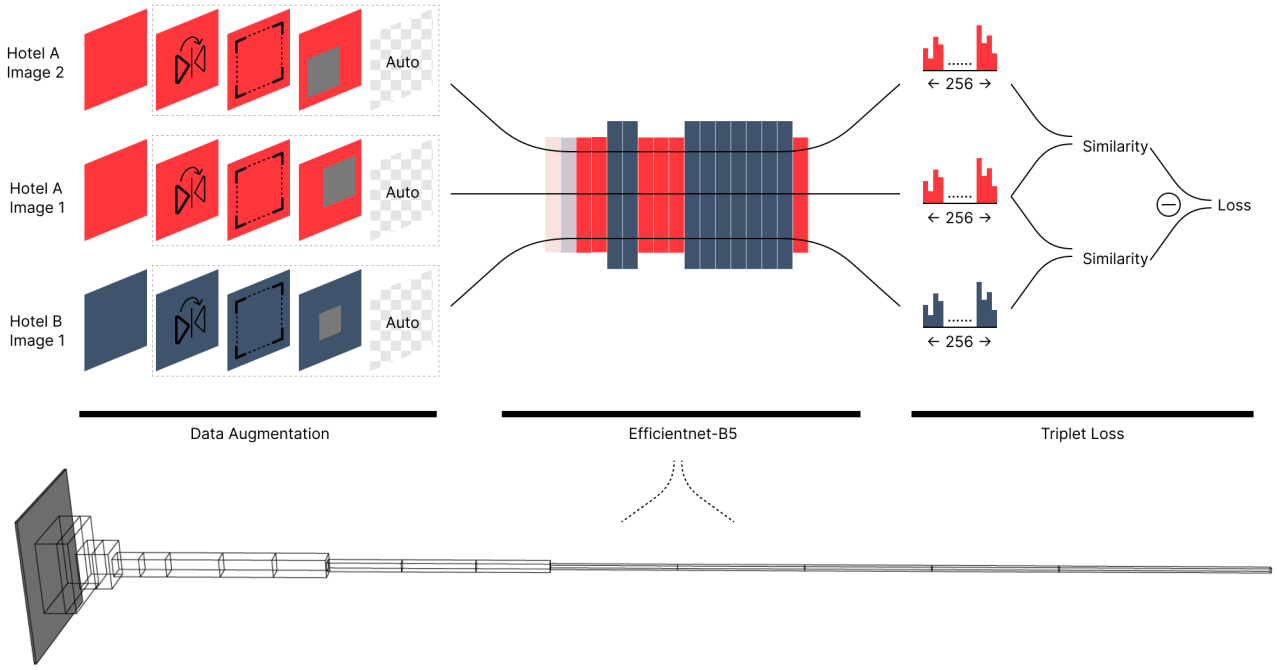


Figure 2. Triplet loss training. Three images, an anchor image, a positive image taken in the same hotel as the anchor image, and a negative image taken in a hotel different from the anchor image, are fed into the neural network to extract their features. The triplet loss encourages a higher similarity between the anchor image and the positive image and a lower similarity between the anchor image and the negative image.

of images and hotel IDs (I, y), where I is a variable-sized RGB image and y is the corresponding hotel ID. The dataset is then split into training set, validation set, and test set. This setting implies that the training set must cover all the hotel IDs in $\mathcal{Y}$; otherwise, there is no means for the model to know the ID of a hotel unseen in the training process.

As the ultimate goal is to assist human trafficking investigations and rescue victims, we are interested in not only one hotel ID but also a short list of probable hotel IDs for any test image so that law enforcement agencies can check a few hotels to rescue victims. Therefore, we extend the model that it predicts not only a single hotel ID for each image but also a ranking of hotel IDs, which was implemented by ranking logits of the classifier.

3.2. Evaluation Metric

We choose the Mean Average Precision at 5 (MAP@5) as the evaluation metric for our task. MAP@5 is defined as

$$\text{MAP@5} = \frac{1}{U}\sum_{u=1}^{U}\sum_{k=1}^{\min(n,5)} P(k) \times \text{rel}(k)$$

where U is the number of images, $P(k)$ is the precision at cutoff k, n is the number of predictions per image, and $\text{rel}(k)$ is an indicator function equaling 1 if the item at rank k is a relevant correct label, zero otherwise. For our case, because there is only one correct answer (the hotel in which an image is taken is unique), the metric can be simplified as

$$\text{MAP@5} = \frac{1}{U}\sum_{u=1}^{U}\frac{1}{R(u)} I[R(u) \leq 5]$$

where $R(u)$ is the ranking of the correct hotel ID and $I[R(u) \leq 5]$ is a binary function that equals 1 when $R(u) \leq 5$.

Besides the quantitative metric MAP@5, hotel images retrieved by the model will also be visualized to analyze which visual features the trained model most relies on.

3.3. Baseline Classifier

We started with a simple classifier, a linear layer on top of a backbone CNN, to classify hotel room images to hotel IDs, i.e.,

$$\text{Prob}(y|I) = \text{softmax}(W \cdot \text{backbone}(I)),$$

where I is the image to be classified, W is the weight matrix of the linear layer, and $\text{Prob}(y|I)$ is the predicted probability that the image is taken at the hotel with ID y. The model is trained with cross-entropy loss on the training set $\{(I_i, y_i)\}_{i=1}^{N}$.

$$L_{\text{CE}} = \frac{1}{N}\sum_{i=1}^{N} -\log \text{Prob}(y_i|I_i).$$

3.4. Backbone and Resolution

We first tried different backbones with various image resolutions to find a good start point. We observed that the resolution of input images is crucial to the problem. Higher image resolution leads to significant improvement of the MAP@5 score. This is reasonable because hotel rooms look similar and hotel recognition relies on details of the hotel rooms. These details are filtered out when the hotel room images are down-sampled to a low resolution. However, higher resolution and larger backbone require more computational cost. Balancing the performance of the model and our budget, we decided to use EfficientNet-B5 [20] through the rest of the project.

3.5. Data Augmentation

Data augmentation is crucial for this problem for two reasons. (1) Images related to human trafficking are usually in low-quality and parts of them are occluded. Data augmentation is necessary to simulate these noises. (2) The available dataset for training is small. Data augmentation could help mitigate this issue. We applied the following data augmentation methods to our models.

- **Random crop** and **random horizontal flip**. These are common methods to augment images. They fit the hotel recognition task because the hotel ID should not be affected by applying these augmentation operations to the images.

- **Random erasing**. We randomly occluded parts of the images to simulate the occlusions appear in real-world human-trafficking images. The occlusions are implemented by randomly erasing rectangular regions of the images.

- **AutoAugment**. Automatic augmentation [2, 3, 12] combines various augmentation methods, e.g. rotating, flipping, translating, shearing, blurring, and color shifting. It finds a good strategy to apply these augmentation methods by searching various strategies on a dataset like ImageNet and comparing the performances of the models trained with different strategies. We tried and compared various automatic augmentation methods in the experiments.

We have already applied random occlusions, random crop, and random horizontal flip to the baseline model when comparing various backbones and resolutions, because the models severely overfit if we did not do so.

3.6. Triplet Loss Training

Besides being formulated as a classification task, the problem can also be approached by learning a metric measuring the similarity of images, such that two images from the same hotel should be similar under this metric. One way to learn such metric is minimizing the triplet loss [17], which is illustrated in Figure 2 and described as follows.

The distance of two images I_1 and I_2 is modeled as

$$\text{dis}(I_1, I_2) = f(\text{backbone}(I_1), \text{backbone}(I_2)),$$

where $f(\cdot,\cdot)$ is a function that measures the (negative) similarity of two vectors; an example would be the Euclidean distance $f(v_1, v_2) = \|v_1 - v_2\|^2$. We randomly sample 3-tuples of images $\{(I_i^{(A)}, I_i^{(P)}, I_i^{(N)})\}_{i=1}^{N}$ from the training set, such that the anchor image $I_i^{(A)}$ and the positive image $I_i^{(P)}$ are taken from the same hotel while the anchor image

$I_i^{(A)}$ and the negative image $I_i^{(N)}$ are not. The triplet loss is

$$L_{\text{triplet}} = \frac{1}{N}\sum_{i=1}^{N}[\text{dis}(I_i^{(A)}, I_i^{(P)}) - \text{dis}(I_i^{(A)}, I_i^{(N)}) + 1]_+$$

where $[x]_+ = \max(x, 0)$.

Minimizing the triplet loss increases the similarity between image features from the same hotel and reduces the similarity between image features from different hotels. Therefore, images from the same hotel will cluster together in the feature space and the clusters of image features from different hotels have a clear gap among them. Features learned this way are expected to be helpful to the classification task.

The triplet loss can either be used as an auxiliary loss or be used alone. When used as an auxiliary loss, the images are still classified by the softmax classifier on top of the CNN backbone, and the model is trained in the way illustrated in the Figure 1. When the triplet loss is used alone, however, the softmax classifier is not trained, so we must rely on the similarity function sim to recognize hotels. Specifically, for any image I in the test set, we find the image I_r in a database that minimizes $\text{dis}(I, I_r)$ and outputs its hotel ID y_r as the predicted hotel ID of the image I.

$$f_{\text{triplet}}(I) = y[\underset{r}{\text{argmin}}\, \text{dis}(I, I_r)].$$

In our case, the database is the training set; for more realistic settings, the database can be updated with newly collected data such that the model can recognize new hotels without retraining. We tested both usages of triplet loss in the experiment section, and found triplet loss as an auxiliary loss was better for the Hotel-ID 2022 dataset empirically, so we used it as our final model.

3.7. Implementation

We used the pretrained backbone CNNs provided by PyTorch [14] and some boilerplate codes in its examples[5]. The code for triplet loss training, nearest-neighbor inference of deep metric learning, strategies for generating random masks, saliency map are implemented by us. We utilized scikit-learn [15] to generate t-SNE.

4. Dataset

We conducted experiments on the dataset provided by the 2022 Hotel-ID to Combat Human Trafficking Competition[6] (Hotel-ID 2022 dataset). Due to limited computational resources available to us, we do not use other datasets in this work, but we are interested in utilizing larger datasets, like Hotels-50K [19], for future work.

[5]https://github.com/pytorch/examples

[6]https://www.kaggle.com/competitions/hotel-id-to-combat-human-trafficking-2022-fgvc9/data

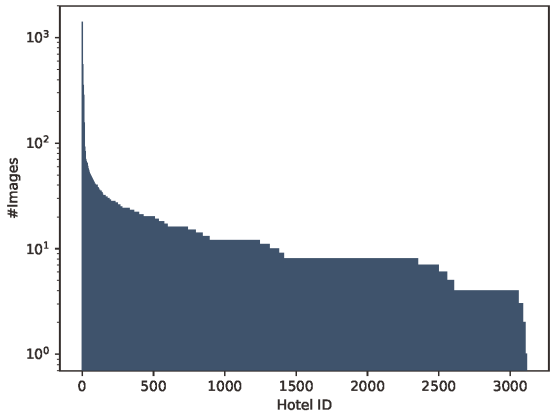


Figure 3. Statistics of the training split of the Hotel-ID 2022 dataset. Note that the y-axis is in log scale. It shows that the problem is unbalanced – a small number of hotels have lots of images, while others have a few images.

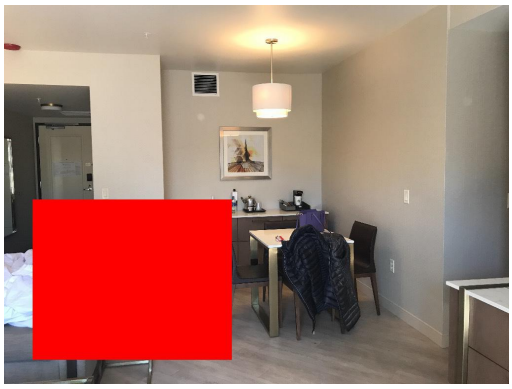


Figure 4. An example image from the Hotel-ID 2022 dataset. It is a photo of a hotel room. A part of the image is occluded by a red block, to simulate that humans in the image may block this part of the room

The dataset consists of crowd-sourced images from the TraffickCam mobile app [18]. TraffickCam mobile app was publicly released for iOS and Android in June 2016. Users around the world can anonymously upload their GPS location, hotel name, room number and take up to four images of their hotel room and bathroom. In fact, photos taken by cellphone cameras tend to be more representative of the type of photos of sex trafficking victims advertised online.

The dataset is split into three parts, training, validation and testing. The training set consists of 44,703 images from 3,116 hotels; the validation set consists of 1,750 images from these hotels; and the test set consists of 3,250 images. As shown in Figure 3, the dataset is unbalanced and has a long tail. Several hotels have lots of images associated with them, among them one hotel has 1393 images, while most

model	#parameters	resolution	MAP@5 ↑
ResNet-152	602 M	224	0.229
ConvNeXt-L	198M	224	0.226
ConvNeXt-L	198M	384	0.273
EfficientNet-B4	194M	224	0.331
EfficientNet-B5	304M	224	0.332
EfficientNet-B4	194M	380	0.359
EfficientNet-B5	304M	456	0.368

Table 1. Comparison of various backbones and resolutions. The resolution of images is crucial to the precision of the model, while model size is not so important compared to the resolution. A higher resolution leads to a better MAP@5 score.

augmentation method	MAP@5 ↑
no augmentation	0.100
random erasing (strategy 1)	0.429
random erasing (strategy 2)	0.437
RandAugment [3]	0.455
AutoAugment [2]	0.461
TrivialAugmentWide [12]	0.466

Table 2. Comparison of various data augmentation methods. The first line shows that the model performs bad without data augmentation. We applied two strategies to augment the images with random erasing (details of the strategies are described in the Section 5.2), and three automatic augmentation methods. The results show that data augmentation boosts the performance of the models significantly.

hotels have around 10 images. Therefore, training a classifier on this dataset is quite hard and can be regarded as few-shot learning to some extent.

The raw images from TraffickCam provide a clear view of hotel images, but there are usually people in images related to human trafficking, which may occlude parts of the hotel room. To simulate this occlusion in real-world scenario, the authors of the Hotel-ID 2022 dataset applied masks to the images randomly, as shown in Figure 4. They applied masks to the validation set and the test set only, and left the training set untouched, providing more flexibility for users of the dataset.

5. Experiments

We conducted experiments on the Hotel-ID 2022 dataset to evaluate our model empirically and analyze various decision choices we made to our model. We also visualized the model with saliency map and t-SNE to evaluate it qualitatively. In this section we also discuss the limitation of our model and propose potential improvements to it that might provide insights to future works.

5.1. Backbone and Resolution

We first compared various backbone CNNs through experiments, as well as the resolutions of the input images. The backbones CNNs we tested include ResNet, ConvNeXt, and EfficientNet, which were all pretrained on ImageNet. Noticing that ConvNeXt and EfficientNet were pretrained with resolutions larger than 224x224, e.g. 384x384, we tried to apply them on 224x224 images and see how much performance they would drop. The experiment result is shown in Table 1. Comparing various backbone architectures, we found that EfficientNet is the best one for our problem, as it achieved the best MAP@5 performance with fewer trainable parameters. As for resolutions, the experiments showed that reducing the resolutions hurt the performances of ConvNeXt and EffcientNet dramatically, so we stuck with high-resolution images in the following experiments. One possible reason that hotel classification requires high-resolution images may be that hotel rooms are quite similar except details like furniture and carpets. These details may be blurred out in downsampled low-resolution images. In the following parts of the experiments, we used EfficientNet-B5 with resolution 456x456.

Optimization and hyperparameters After choosing the backbone and the resolution, we tuned the optimizer and hyperparameters. Starting with SGD without weight decay, we first replaced SGD with AdamW. Then, we applied cosine annealing to schedule the learning rate, that starts at 0.001 and becomes zero after 90 epochs. Finally, we spent the first 5 epochs to linearly warm up the learning rate from zero. These strategies significantly improved the model performance from 0.368 to 0.429.

5.2. Data Augmentation

We compared various data augmentation methods in this section. As described in the Section 4, the red occlusion masks were only applied to the validation set and the test set but not the training set in the Hotel-ID 2022 dataset. Therefore, not using any augmentation method will result in a mismatch between the training stage and the inference stage. Because processing images without occlusions during training is much easier than processing images with occlusions on inference, training the model without any data augmentation method resulted in a very bad performance, as shown in the first line of Table 2.

For adding occlusions to the training images, we tried two strategies to randomly mask parts of the images. The first strategy is the `RandomErasing` class in PyTorch, which randomly samples a rectangle and erases the context in it. The second strategy is to mimic the strategy used by the dataset authors to create the dataset. The dataset comes

training	inference	MAP@5 ↑
classif.	classif.	0.466
metric	metric	0.179
metric + classif.	classif.	0.473

Table 3. Comparison of two approaches to the problem, as image classification (classif.) and as metric learning (metric).

with around 5,000 images that each illustrates an area of the image that is masked, we believe these masks are generated with the same distribution as they added occlusions to the images in the validation set and the test set. For each training image, we randomly sampled a mask from them and applied it to the image. A comparison of these two strategies are listed in the second line and the third line of the Table 2. It shows that the second strategy, i.e., applying the same mask strategy as the validation set and the test set, is better. This is expected, because training on the images with the same distribution as the images the test set generally reduces the generalization gap.

In addition to random erasing, we applied several automatic augmentation methods and compared their performances, which are listed in the last three lines of the Table 2. We did not remove random erasing described in the previous paragraph, although automatic augmentation is usually more powerful than random erasing for general datasets. The reason behind it is that random erasing not only augments the training set, but also reduces the discrepancy between the training set and the test set.

5.3. Triplet Loss Training

The hotel room classification problem can also be approached as learning a metric that measures the similarity of images. In this section, we implemented this approach and compared it to the image classification approach implemented in the previous sections. Specifically, we compared three approaches.

1. Using the usual image classification technique, i.e., training the model with cross entropy error and ranking hotel IDs by their logits on inference.

2. Learning a metric of images using triplet loss, and classifying the images using a nearest-neighbor classifier based on this metric. Since we measured the performance of the model by MAP@5 which requires 5 hotel IDs rather than one, we did not apply KNN, but instead ranked the neighbors to get a list of hotel IDs.

3. We trained an additional linear classifier on the image features learned in the previous approach with cross entropy loss, and inferred the hotel IDs in the same way as the first approach.

The experiment results are listed in the Table 3. We found that the second approach, metric learning, did not perform as well as the first approach, image classification. Image classification is a more direct formulation of the problem than metric learning, so it was expected that the first approach got a better performance. But we should also notice that metric learning got a non-trivial result, and it generalized better to real-world settings as the set of hotels is usually not provided during the training stage in real-world applications. Moreover, augmenting image classification with metric learning as an auxiliary task improved the performance a little.

5.4. Metric on Test Set

Finally, we tested our best model on the test set, it achieved an MAP@5 of 0.454 (0.473 on the validation set). We also tried ensemble of models because the training set is rather small. An ensemble of 7 models reached 0.510 MAP@5 on the validation set and 0.488 MAP@5 on the test set. The results indicated that our model also performed well on the test set and did not overfit on the validation set.

5.5. Visualization

Saliency map. We analyzed the saliency maps (Figure 5) of our model. The saliency maps can reveal which pixels in the image have a significant impact on the model results. From the results of the saliency maps, we find that the items and furniture types in the hotel rooms are almost the same. Therefore, less often diverse furniture is not important in interior classification, for example, beds and TVs in different hotels often look the same and thus are not important. In contrast, the textures and materials of each surface, such as walls, curtains, and vanities, are significant in the results. In addition, lamps, and sometimes tables and chairs, often have unique shapes related to the style and budget of a hotel chain, which can be helpful when they appear in the image. Sometimes, special colors and decorators help too. As a sanity check, we find that the pixels of the masks have no impact on the model results, which proves that the model successfully learned to ignore these masks and focused on the informative parts of the images.

t-SNE. We visualized the features learned by the models with t-SNE. t-SNE is a tool that maps high-dimensional data into low-dimensional data (2D for visualization purpose) while trying its best to maintain distances between data points. If the model learns good features of images, the features of the images of the same hotel will cluster together and features across different hotels will separate apart. Because there are over three thousand hotels in the dataset, visualizing all of them will lead to a messy figure. Therefore, we chose 20 hotels from the dataset and visualized

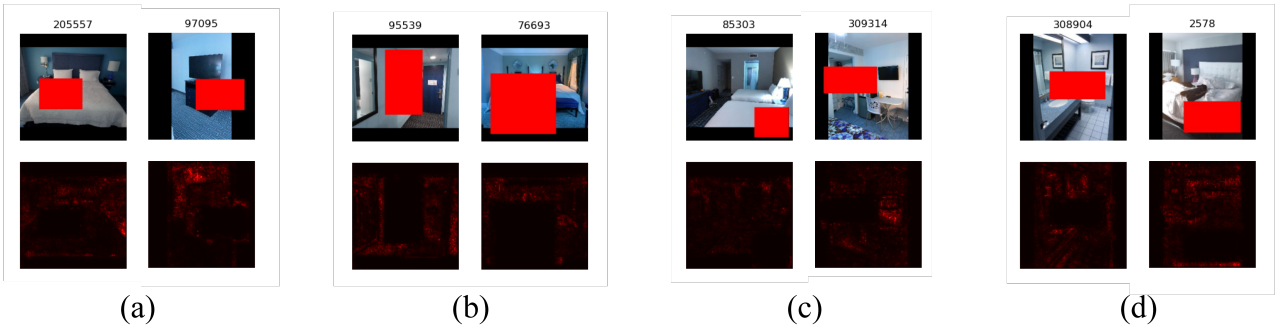
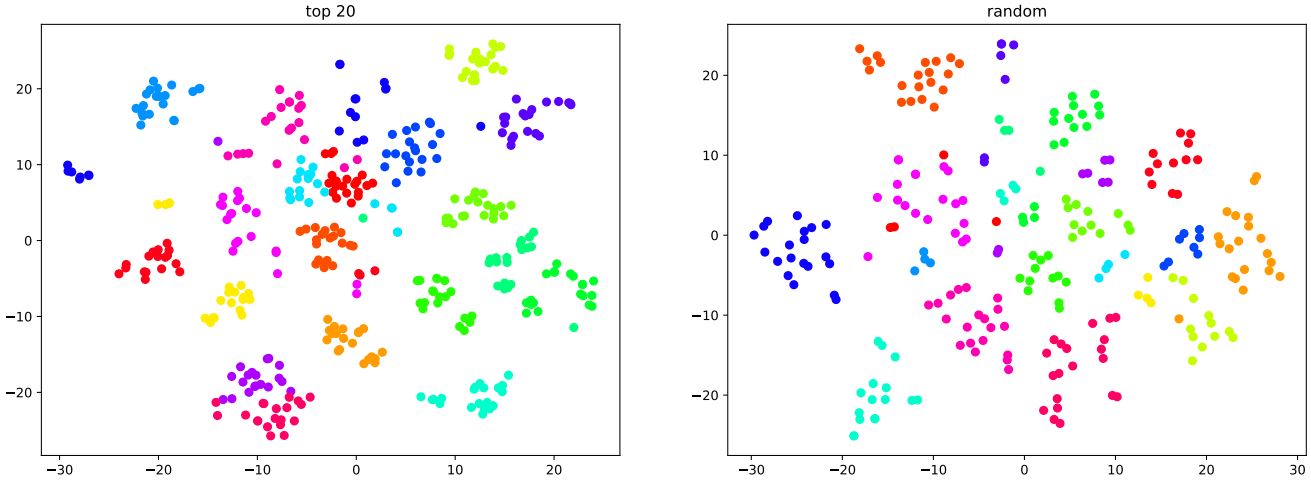


Figure 5. Saliency Map: (a) Common furniture is not important; (b) Special colors (door of the left one) and decorators (wall decorator of the right one) matter; (c) Distinction shapes (desk lamp of the left one and chair of the right one) help; (d) Textures and materials are the most common factors for identification.



(a) Top 20 hotels that have the most number of training images.

(b) Random 20 hotels.

Figure 6. t-SNE of image features for 20 hotels. Each dot stands for an image and each color represents a hotel.

them. The set of 20 hotels were determined by two strategies. (1) We selected the top-20 hotels that had the most number of images in the training set. (2) We randomly sampled 20 hotels. Figure 6a and Figure 6b are t-SNE visualization with these two sets of hotels, respectively. Each dot in the figures stands for an image and each color represents a hotel. These figures showed that images from the same hotel were basically clustered together and there was some distance between different hotels. We also found that the results for top-20 hotels was better than 20 random hotels, as the clusters of images were more compact for the former ones than the latter ones. This result was expected, because top-20 hotels had more training data. It also implied that lack of training data was a major issue of the problem and collecting more data would be a promising direction for future works.

6. Conclusions

We proposed a system for hotel recognition, which is helpful for combating human trafficking. The system classifies hotel room images with EfficientNet on high-resolution inputs. The neural network was trained with both image classification task and deep metric learning task. Data augmentation is crucial to our system, and we utilized random erasing and automatic augmentation. Quantitative evaluation on the Hotel-ID 2022 dataset showed that our system effectively recognized hotels; qualitative visualizations implied that the neural network extracted useful features from hotel room images. We suggested that collecting more data is a promising direction for future works.

Contributions

Peng Chen implemented the system, designed and ran the experiments. Yuyu Lin visualized the model and plotted illustration figures of the system. Chi On Ho implemented random erasing and surveyed previous works on hotel recognition and related areas. We referred examples of PyTorch (https://github.com/pytorch/examples) for boilerplate codes.

References

[1] Hamidreza Alvari, Paulo Shakarian, and J. E. Kelly Snyder. Semi-supervised learning for detecting human trafficking. *CoRR*, abs/1705.10786, 2017. 2

[2] Ekin D. Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 113–123. Computer Vision Foundation / IEEE, 2019. 1, 4, 6

[3] Ekin Dogus Cubuk, Barret Zoph, Jonathon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 1, 4, 6

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

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

[6] Weifeng Ge. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–285, 2018. 2

[7] Munawar Hayat, Salman H Khan, Mohammed Bennamoun, and Senjian An. A spatial layout and scale invariant feature representation for indoor scene classification. *IEEE Transactions on Image Processing*, 25(10):4829–4841, 2016. 2

[8] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *CoRR*, abs/1703.07737, 2017. 2

[9] Donna M Hughes. The use of new communications and information technologies for sexual exploitation of women and children. *Hastings Women's LJ*, 13:127, 2002. 2

[10] Rashmi Kamath, Gregory Rolwes, Samuel Black, and Abby Stylianou. The 2021 hotel-id to combat human trafficking competition dataset. *CoRR*, abs/2106.05746, 2021. 1, 2

[11] Yabei Li, Zhang Zhang, Yanhua Cheng, Liang Wang, and Tieniu Tan. Mapnet: Multi-modal attentive pooling network for rgb-d indoor scene classification. *Pattern Recognition*, 90:436–449, 2019. 2

[12] Samuel G. Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 754–762. IEEE, 2021. 1, 4, 6

[13] Megha Pandey and Svetlana Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *2011 International Conference on Computer Vision*, pages 1307–1314, 2011. 2

[14] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 5

[16] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE conference on computer vision and pattern recognition*, pages 413–420. IEEE, 2009. 2

[17] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 815–823. IEEE Computer Society, 2015. 2, 4

[18] Abby Stylianou, Jessica Schreier, Richard Souvenir, and Robert Pless. Traffickcam: Crowdsourced and computer vision based approaches to fighting sex trafficking. In *2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–8, 2017. 5

[19] Abby Stylianou, Hong Xuan, Maya Shende, Jonathan Brandt, Richard Souvenir, and Robert Pless. Hotels-50k: A global hotel recognition dataset. *CoRR*, abs/1901.11397, 2019. 2, 5

[20] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 2019. 4

[21] Daniel Sáez Trigueros, Li Meng, and Margaret Hartnett. Enhancing convolutional neural networks for face recognition with occlusion maps and batch triplet loss. *Image and Vision Computing*, 79:99–108, 2018. 2

[22] Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10:207–244, jun 2009. 2

[23] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*

2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 13001–13008. AAAI Press, 2020. 1