RIPA: Real-time Image Privacy Alert System

Dakshak Keerthi Chandra
Computer Science Department
Missouri University of Science and Technology
Rolla, MO, USA
dkthc@mst.edu

Weerdhawal Chowgule
Computer Science Department
Missouri University of Science and Technology
Rolla, MO, USA
wcmb3@mst.edu

Yanjie Fu
Computer Science Department
Missouri University of Science and Technology
Rolla, MO, USA
fuyan@mst.edu

Dan Lin *
EECS
University of Missouri
Columbia, Mo, USA
lindan@missouri.edu

***Abstract*—The problem of privacy and security threats arising from images uploaded onto popular social media and content sharing websites is prevalent now more than ever. As our digital footprints grow exponentially, the need to find a solution to these problems has become that much more significant. In order to address these problems, a lot of research work has been carried out for image privacy protection through privacy policy recommendations and configurations. Due to the recent advancement in the field of computer vision and deep learning we can now gain more detailed insights about the context of an image and about the relationships between objects within it, this makes it possible to better address these problems.**

The privacy and security threats arising from an image uploaded on-line are not only limited to the data owners. Unlike previous works that are mostly focused on individual privacy policies, we take into account privacy concerns of multiple objects depicted on the same photo (even people, animals or other objects in the background of a scenery photo) whereby these privacy concerns may not be those from the user who uploads the photo. Specifically, we first build a general knowledge base by leveraging convolution neural networks to classify sensitive and non-sensitive image content and then use our proposed metadata analysis module to analyze metadata embedded within the image. Next, we extract objects present in the photo and validate if there is any privacy violation of the objects' privacy concerns. If any sensitive object is found, we toggle the object and issue a privacy violation alert to the user who is uploading the image as well as the service provider.

[1] ***Keywords*—Image privacy, Geo-location, Convolution Neural Network, Transfer Learning**

I. Introduction

With the occurrence of any significant event in today's smart world, the word is propagated everywhere within no time through the Internet. The mode of broadcast can vary from text, voice, image to video. Due to the pervasive use of cellular cameras, digital cameras and also because of the escalation in number of content sharing and social networking sites such as Facebook, Twitter, Instagram, Linkedin, Pintrest etc., and the ease of access they provide for an user to upload and share images on-line, the risk of a personal image of an individual or an object within it being accessed and misused by an adversary becomes a growing concern.

Further, the privacy and security threats arising from an image uploaded on-line are not only limited to the photo owners. An image, apart from its visual context, can also contain a lot of sensitive information embedded within itself. Metadata like time-stamp and geo-tag embedded within an image can contain sensitive and vital information pertaining to the people or objects within it, about which the user uploading the image might be unaware of and disclose it. This can lead to a lot of potential privacy and security threats that have not been addressed in the existing works. Specifically, for a group photo, different people in the photo may have different privacy concerns and some of them may not want the photo to be shared publicly. For example, one may upload images of his/herself in a pub with lots of other people in the background. The photo owner may have taken certain precautions like setting up privacy filters that decides who may or may not view the image to protect his/her privacy before he/she uploads the photo, but by doing so, the user might unknowingly be endangering the privacy of other people in the background of the image who might not wish to be photographed. Even people in the background of a scenery image may have their own privacy concerns. For example, they may not be willing to disclose their locations because of the scenery images uploaded by others.

The privacy concerns are further extended to other objects in the photos. There may be sensitive objects in the photo, such as endangered animal species and military items. Sharing images containing such sensitive objects may reveal the locations of these objects and raise threats to these objects. For example, a user uploading pictures of himself on a safari with an endangered species in the background, while thinking that he is just uploading his vacation photo. If the image is uploaded with its geo-location metadata intact on social media or content sharing websites, there is a chance of exposing the location of the endangered species to the poachers who may use this information to go hunt it down.

Not everyone is aware of the repercussions of images they upload on-line and privacy and security threats they pose. A

[1] *: contact author

user uploading an image may unknowingly be compromising privacy and security of another person or object within the image. This is why there is a need for a system that can detect sensitive objects within images in real time and warn the users of any repercussions that may occur if the image is uploaded on-line. In order to prevent the privacy breach from the aforementioned cases, we propose a system called RIPA (Real-time Image Privacy Alert System). Unlike previous studies [1]–[3] that are mostly focused on individual privacy policies, we take into account privacy concerns of multiple objects depicted on the same photo whereby these privacy concerns may not be only those from the user who uploads the photo. Specifically, we first build a general knowledge base by leveraging convolution neural networks to learn sensitive and non-sensitive image content and then use our proposed metadata analysis module to analyze metadata embedded within the image. Next, we extract objects within a photo and validate if there is any privacy violation of the objects' privacy concerns. If any sensitive object is found, we toggle the object and issue a privacy violation alert to the user who is uploading the image as well as the service provider. Specifically, our RIPA system is composed of the following three major modules: Convolution Neural Network for image classification, metadata analysis and Policy Alert Issuance module and the RIPA data storage system module.

- *Convolution Neural Network:* To account for image content, we propose two multi-layered convolution neural network frameworks. The main CNN is implemented using Transfer Learning technique and the other CNN is specifically trained from scratch for facial recognition. The RIPA system leverages advanced image classification techniques to classify 81 different classes of images. We perform supervised learning which uses the spectral signatures obtained from training samples to classify an image.
- *Metadata Analysis and Policy Alert Issuance:* The images classified as sensitive then go through the metadata analysis module, which checks for any sensitive metadata associated with it. If any sensitive data that can give rise to privacy and security threats are detected, a policy alert is issued based on it. The metadata analysis module calculates a security feature depending on which class an image is classified (this is explained in detail in the later sections).
- *RIPA Data Storage System:* RIPA system stores sensitive location information and user data. The data storage system is optimally structured to reduce the time taken to perform data lookup operations as these kinds of operations in huge databases can cause significant overhead.

The remaining of the paper is organized as follows. Section II reviews previous related works that influenced our work. Section III presents the detailed algorithm of our proposed system. Section IV reports performance evaluation results. Finally, Section V concludes the paper and outlines the future work.

II. Related Work

In this section, we first review existing works on image privacy, and then discuss the applications of deep learning since it is one of our building blocks.

A. Image Privacy

Due to the privacy and security concerns posed by content sharing and social media websites to their users there is a growing interest for research in this area. However, most of existing works mainly focus on privacy protection for photo owners and co-owners. Little work has been done to provide privacy protection for other sensitive objects in the background of the photos, and little has explored the location and time based privacy and security threats users are exposed to due to the photo sharing on social networking sites.

One representative work on image privacy is by Squicciarian et al., called A3P [1]. The authors of this paper have used a dataset which contained five generic image classes and each of these classes contained an approximate of 100 images. Content-based classification was implemented using Jacob algorithm [4]to construct an efficient classifier which assigns a class to each image, also it generates image signatures. Whenever a user uploads an image a signature is generated and is compared with the existing signatures of images in the current image database and set to the class which has the most similar signature, but just in case there are no matches another new class is generated. To verify the accuracy of their model they tested against a popular classified data-set, image- net.org [5] and the lowest accuracy was 86.4% for the class scene and highest being 100% for kids. It is composed of a large number of highly interconnected processing elements(neurons) working in unison to solve specific problems. However, this work mainly consider the privacy of the photo owners, whereas our work provides a broader privacy protection for any sensitive objects in the photo.

Another related work is by Fan et al. [2] who leverages deep learning techniques to better classify sensitive images in order to generate privacy policies mainly for photo owners. The basic idea is detection of privacy-sensitive objects automatically for the images being shared by recognizing classes and identifying their privacy settings. Once the settings are identified the owner can we warned about the objects needed to be protected before sharing. The authors also developed a hierarchical deep multi-task learning (HD-MTL) algorithm to learn more representative deep CNNs and more discriminative tree classifier jointly over the visual tree, achieving fast and accurate detection of large number of privacy-sensitive object classes. And to enhance the performance of the hierarchical object detection by exploiting multiple paths simultaneously a soft prediction is used. Finally achieving image privacy protection by blurring the privacy-sensitive objects automatically.

Other works that are related to ours are the following. Ahern et al. [3] analyzed effectiveness of tags as well as location information in predicting privacy settings of the photos. Works by Ames et al. [6], Miller et al. [7] and Besmer et al. [8] also explored relationship between images and tags associated with

them and how they are often used to communicate contextual or social information with those viewing the photo. In addition, several interesting works have been done to automate the privacy settings in-order to limit the sensitive data of users from being exposed in social networking sites and to protect the privacy of the users [9]–[13]. Ravichandran et al. [14] studied the diverse nature of people's privacy preferences. The research was conducted in the context of location-sharing applications, where they studied how to predict a user's privacy preferences for location-based data based on location and time of day.

Our work also shares some similarities with some recommender systems. Chen et al. [15] proposed a recommendation framework based on learning latent space representation of the groups which can then be used to recommend the most likely groups for a given image shared in online social medial which can help connect image content with communities. They characterize images through three types of features: visual features, user generated text tags, and social interaction, based on which they recommend the most likely groups for a given image. [16] proposed a system called SheepDog which automatically adds photos into appropriate groups and recommends suitable tags for users on Flickr. They utilize concept detection to predict relevant concepts of a photo and probe into the issue about training data collection for concept classification. By utilizing preexisting information from Flickr, they implement a rank based method to obtain reliable training data and provide reasonable group/tag recommendations for input photos.

B. Deep Learning

Some of the work in computer vision and deep learning that also influenced work are as discussed. Alex Krizhevsky [17] et. al proposed a model that was used to classify around 1.2 million images in 1000 different classes. This model contains 5 convolutional layers, followed by some max pool layers and 3 fully connected layers. To perform such huge classification proper dataset is also required and the largest and the best datasets available are LabelMe [18] consists of hundreds of thousands of fully-segmented images and ImageNet [5], which consists of over 15 million labeled high-resolution images in over 22,000 categories. One of the points to note from this model is that a deep convolution neural network can achieve very good results by using supervised learning.

Hastie et al. [19] proposed Supervised learning. It is a machine leaning task of inferring a function from supervised training data. A supervised learning algorithm analyzes the training data and produces an inferred function, which is called a classifier (if the output is discrete) or a regression function (if the output is continuous). The inferred function predicts the correct output value for any valid input object.

Yi Sun [20] et al. proposed DeepID3 for face recognition which are two very deep neural network architectures that are rebuilt from stacked convolution and inception layers proposed in VGG net and Google net that are suitable for face recognition. DeepID3 net1 contains eight continuous convolutional layers with four pooling layer after each pair where as DeepID3 net2 starts with four convolutional layers with two pooling layers after each pair, followed by a pooling layer which has three inception layers before it and two after ending with the last pooling layer. The idea behind the stacking convolution/inception layers before each pooling layer is that there is depth of DeepID3 and helps to form features with larger receptive fields. The pooling layers are fully connected networks which are provided with supervisory signals(aka supervised data) which helps to learn better mid-level features and makes optimization of a very deep neural network easier. Finally after the model was built the last layer would be trained on supervised user data which would be used for classification of the images.

III. System Architecture

In this section, we present our proposed RIPA system which consists of three major modules: (i) image classification module, (ii) metadata analysis and policy alert issuance module; and (iii) the RIPA data storage system.

Figure 1 gives an overview of RIPA system. The data flow in the RIPA is as follows. When a user uploads an image, the image will pass through the main transfer learning model, which then categorizes the image as either belonging to sensitive or non-sensitive class. There are two main types of sensitive classes: (i) human subject class and (ii) non-human subject classes such as endangered animals and security sensitive items. If the image is classified as human subject class, it will be further sent to the facial recognition module which combines the Histogram of Oriented Gradients (HOG) methodology and a CNN model. This facial recognition component will generate identities of the people within the image. For either type of the sensitive images, it will be further sent to the second module for metadata analysis. The metadata analysis includes the analysis of privacy policies of users in the image and cross checking of geo-location tags against the sensitive locations flagged by the users. If any potential privacy violation is detected, the affected object will be toggled in the image, and an alert will be issued to the image owner as well as the service provider. The third module in the system is in charge of data management to speed up the whole analysis process.

In what follows, we provide more technical details of each module.

A. Convolution Neural Network for Image Classification

We leverage Convolution Neural Network (CNN) for image classification since it works phenomenally for this kind of task and somewhat mimics the Human Visual System [21]. Specifically, we make use of convolution and rely upon max-pooling and FC layers to reduce the image size and retain the information. We employ two CNN's. One is based on a transfer learning model called MobileNet [22] which is used as our main model. The other is a new CNN for facial recognition. Both of our CNN models were built and trained using TensorFlow [23].

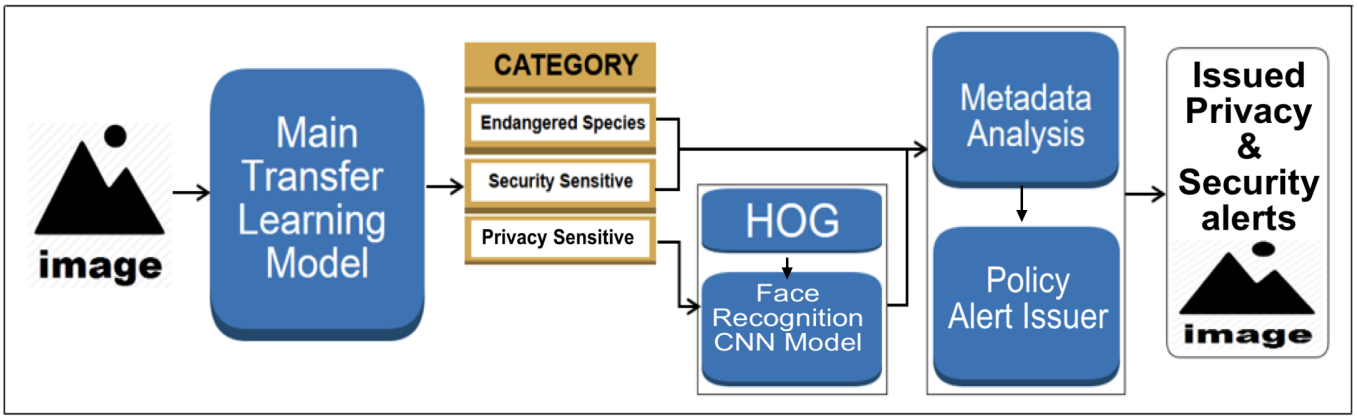

Fig. 1. Framework of the proposed RIPA model

- *Construction of the main CNN*: We use the transfer learning technique to leverage the knowledge of pre-trained MobileNet model [24] and then retrain it on our collected RIPA dataset for better image classification accuracy. Transfer Learning [25] is a technique widely used in deep learning, in which knowledge gained by a model while being trained on one type of problem is used to train on a similar type of problem. In our system, since the first few layers of the deep learning model are sufficient to identify features of the problem, we remove the last few layers of the pre-trained model and retrain it on new layers by using different parameters suited for our problem. In this way, we utilize the knowledge gain to make much more accurate predictions.

- *Construction Facial Recognition CNN*: We build and train our own CNN for the purpose of Facial Recognition. For this CNN we use ReLU as our activation function which is denoted by

$$f(x) = max(0, x)$$

We use ReLU because, deep convolutional neural networks with ReLU's train several times faster than their equivalents with tanh units [17]. Logistic and hyperbolic tangent networks suffer from the vanishing gradient problem [26], where the gradient essentially becomes 0 after a certain amount of training (because of the two horizontal asymptotes) and stops all learning in that section of the network. ReLU units are only 0 gradient on one side, which empirically is superior. In our system, max pooling is conducted by applying a max filter to non-overlapping subregions of the initial representation. Max pooling [27] is a sample-based discretization process and its objective is to down-sample an input representation of an image, hidden-layer output matrix, etc., reducing its dimensionality and allowing for assumptions to be made about features contained in the sub-regions binned. This is done in part to control over-fitting by providing an abstracted form of the representation. It reduces the computational cost by reducing the number of parameters

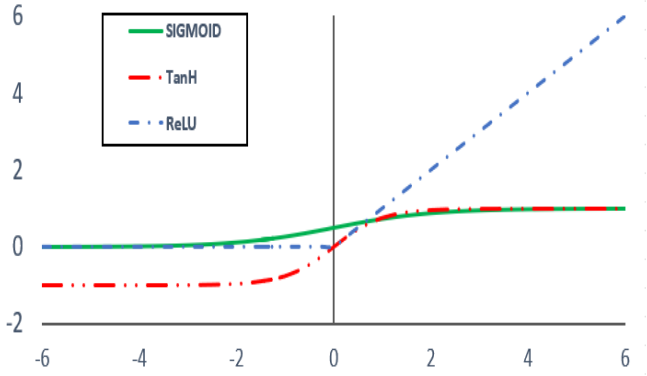

Fig. 2. Activation functions comparison

to learn and provides basic translation invariance to the internal representation.

We also adopt Adam Optimization algorithm [28] as our optimizer as it combines the advantages of two other extensions of stochastic gradient descent algorithms, Adaptive Gradient Algorithm (Adagrad) [28] and Root Mean Square Propagation (RMSProp) [28]. Adam is a popular algorithm in the field of deep learning because it yields good results fast. Empirical results [28] demonstrate that Adam works well in practice and compares favorably to other stochastic optimization methods.

We employ Dropout [29] for regularization. It is a regularization technique for reducing over-fitting in neural networks by preventing complex co-adaptations on training data. It is a very efficient way of performing model averaging with neural networks. The term "dropout" refers to dropping out units (both hidden and visible) in a neural network.

Moreover, we also define Softmax function which can

map a vector to a probability of a given output in binary classification. The Softmax function is defined as

$$h_\theta(x) = \frac{1}{1 + exp(-\theta^T x)}$$

where, θ represents a vector of weights, and (x) is a vector of input values. This function is used to approximate a target function in binary classification. The softmax function produces a scalar output $h_\theta(x) \epsilon \mathbb{R}, 0 < h_\theta(x) < 1$. This can be seen as the confidence that the test point

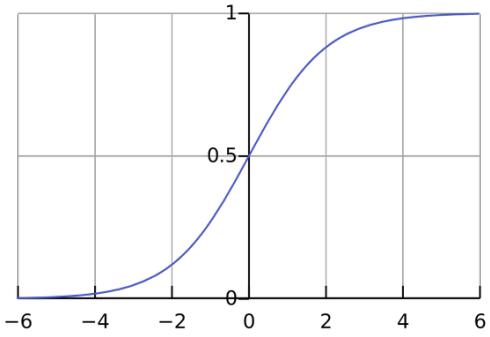

Fig. 3. Softmax function

has an output value of 1. When $-\theta^T x$ is very small, then the probability y=1 is small. When $-\theta^T x$ is very large, $h_\theta(x)$ approaches 1 as the probability that y=1 approaches 100%. Softmax is most widely used because in neural network each neuron receives a vector of outputs from other neurons that fired, each axon with its own weighting. These are then linearly combined and used in the softmax function to determine if the next neuron fires or not.

- *HOG and CNN for face recognition*: Our system performs facial recognition for images that contain human subjects in it. The goal is to identify these human subjects so that the system can verify all the human subjects' privacy policies are not violated. To achieve this, we combine the advantages of HOG methodology [30] and a CNN model trained specifically on our collected dataset crawled using Google API. HOG is a feature descriptor widely used in the field of computer vision for object detection.

  The process of face detection begins by searching for human eyes. Once detected, we then attempt to detect other features of the face such as eyebrows, mouth, nose, nostrils building a template using these set of features. We count occurrences of gradient orientation which is computed on a dense grid of uniformly spaced cells and use overlapping local contrast normalization to improve accuracy. More specifically, the essential thought behind the histogram of oriented gradients descriptor is that the local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. The image is divided into small connected regions called cells. For the pixels within each cell, a histogram of gradient directions is compiled. The descriptor is the concatenation of these histograms. For improved accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block. Then, this value can be used to normalize all cells within the block. This normalization results in better invariance to changes in illumination and shadowing. Separated facial features extracted from the HOG module is then passed through the CNN module specifically trained for facial recognition. For our Facial Recognition CNN model, the learning rate is set to be 0.1 and images of size $32 \times 32 \times 3$ were used.

The image classifier will return the category of the image being uploaded. If the image belongs to sensitive class, it will be forwarded to the following metadata analysis module for more detailed analysis.

B. Metadata Analysis and Policy Alert Issuance

In this module of the RIPA system, it takes the classification result as inputs from either the main Transfer Learning Convolution Neural Network or the Facial Recognition CNN model, and then analyzes the metadata of the images. Depending on the findings of the analysis, a policy alert may be issued when a privacy violation is detected.

Algorithm 1 RIPA Algorithm

Step 1: Classify input image I_m using the main transfer learning model T_m
Step 2: Check if classified Image belongs to sensitive class S_c otherwise perform Step 12
Step 3: If I_m belongs to S_c category check which subclass in S_c it belongs to
Step 4: If I_m belongs to S_c subclasses Endangered Species S_{es} or subclass Security Sensitive S_{ss} perform Step 10
Step 5: If I_m belongs to S_c subclass Privacy Sensitive P_s perform Step 6
Step 6: Identify facial features using HOG, extract individual faces from I_m and perform Step 7
Step 7: Identify individual user identity using Face Recognition model F_{rm} and perform Step 8
Step 8: Cross reference F_{rm} predicted identities with RIPA data storage system R_{ds} for any matches
Step 9: If matches in R_{ds} are found perform Step 10 else perform Step 12
Step 10: Analyze metadata of I_m to check for any sensitive information S_{mtd} embedded within the image
Step 11: If S_{mtd} found issue privacy alerts accordingly else perform Step 12
Step 12: Upload Image

Specifically, if the image belongs to sensitive non-human object category, our system will check if its geo-location tag is intact and contains detailed location information. If so, our system will toggle that object on the image and issues a privacy alert with the explanation of that particular object category.

If the image belongs to human subject category, the process is a little more complex. We will need to check the privacy policy for each identified human subject in the photo. Here, we propose a fine-grained location-aware privacy policy. Each user in the social network is allowed to specify not only to whom they would like to share the photos like most existing social networking sites, but also the sensitive locations if included in the photo that they do not wish to be shared at all.

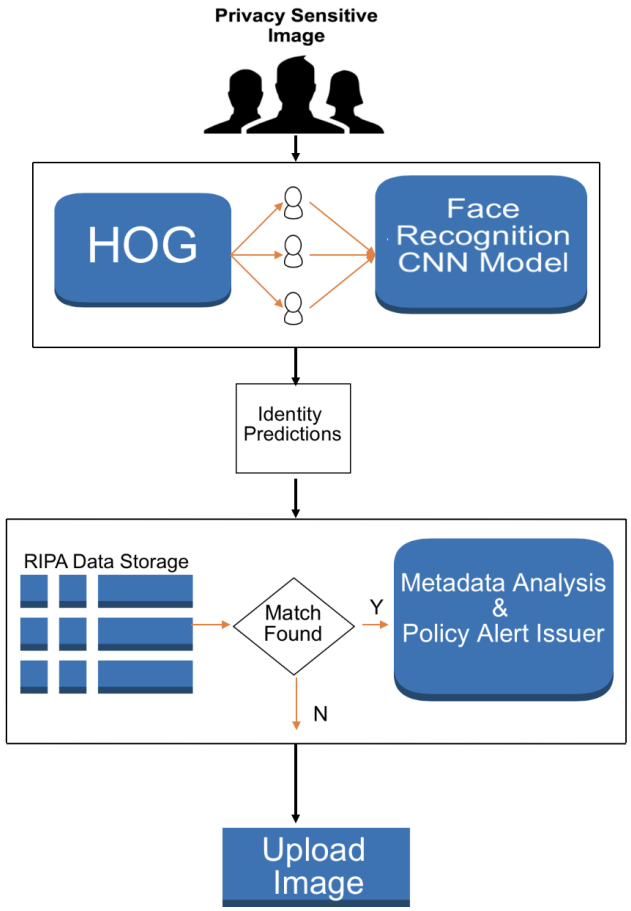

Fig. 4. Framework of Face Recognition module

The next subsection provides more details of the management of such location-aware privacy policies. If a person in the photo has a privacy policy conflicting with the photo owner's sharing plan, we will issue a privacy alert. For example, the photo owner plans to share the photo publicly to everyone in the social network. However, another person (who may be in the background of the photo) has a privacy setting that clearly flags the location of this photo as sensitive, our system will issue a privacy alert to the photo owner that he/she may be violating others' privacy. The same alert message will be forwarded to the service provider for record in case any future dispute regarding privacy violation.

C. RIPA Data Storage System

RIPA uses a PostgreSQL database system to store the location-aware privacy policies whereby user can specify the

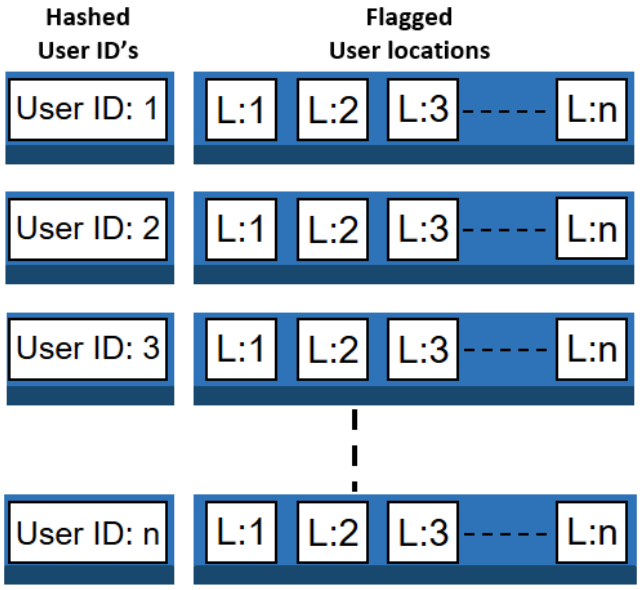

Fig. 5. RIPA Data Storage schema

sensitive locations that they do not want to disclose. The users policies are stored in a hash table. Given a user ID, the list of sensitive locations can be quickly located as shown in Figure 5. The individual flagged location of the users are geo-hashed and stored in a sorted order. We then take advantages of this kind of sorted data set during the search to speed up the process. As the user IDs are stored using hash functions, the name lookup function time complexity will be O(1). We then implement the interpolation search while cross referencing the geo-location found in the metadata of the image with the location flagged by the user. The time complexity for the location search would be O(log log n). The search result is then passed on to the Policy Alert Issuance submodule which then issues alerts accordingly.

IV. Experimental Study

In this section, we give a detailed description of the dataset used, and then report our experimental findings.

A. Dataset Description

Our RIPA dataset consist of 55,462 images in total. It has 81 classes, with each sensitive image class consisting of around 1000 images. Out of the 55,462 images 40,000 images were used as training and testing datasets for CNN. The remaining 12,182 images which had metadata associated with them were crawled using Flickr API, these were solely used to test our metadata analysis and Policy Alert Issuance module.

To test our facial recognition CNN model we also crawled 3280 images of well known personalities using Google API. We crawled images for 8 different identities with each identity consisting of about 400 images each. Of these 2500 images were used to train the model and the remaining 780 images were used for testing.

We further refined the images by dividing them into two main categories: sensitive and non-sensitive. 20 classes were

Sensitive Category	Number of Images
Tiger	1,032
Siberian Crane	1,113
Panda	1,072
Rhino	1,211
Gorilla	1,064
Snow Leopard	1,102
Monarch Butterfly	1,002
Axolot	988
Giant ibis	1,041
Angler Fish	1,122
Elephant	1,087
Hawksbill turtle	1,312
Kiwi	1,053
Sloth	1,064
Whale	1,008
Protests	1,132
Military	1,092
Military Vehicles	1,173
Accidents	1,236
fire	1,041

TABLE I
DATA STATISTICS ON IMAGE CLASSES UNDER SENSITIVE CATEGORY

Identities	Number of Images
Barack Obama	410
Bradley Cooper	412
Brad Pitt	407
Jennifer Lawrence	412
Julia Roberts	402
Meryl Streep	415
Stephen Hawking	411
Tom Cruise	414

TABLE II
DATA STATISTICS ON DIFFERENT IDENTITIES USED FOR TRAINING FACIAL RECOGNITION MODULE

classified under sensitive category and the rest were classified under non-sensitive category. All the images were then resized to $128 \times 128 \times 3$ dimension to be fed as inputs to the main Transfer Learning Convolution Neural Network. Images of size $32 \times 32 \times 3$ were used to train the Facial Recognition CNN module. We mainly focus on the classes of images that come under sensitive category, using these image classes we derive some interesting results.

B. Experimental Results

1) CNN Training Details and Results: At the beginning, we provide the details of the parameters used to train our CNN models along with the obtained results. We trained our main RIPA transfer learning model by setting the learning rate to be 0.01. 81 labels/classes were one-hot encoded and images of size $128 \times 128 \times 3$ were used. For training we used 30,000 images, and for validation we used 10,000 images. The number of training steps were limited to 500. Training and validation phases completed using a GPU took about 30 minutes to finish. On a CPU with 8 cores, it took about 3.5 hours. Figure 6 shows the computational graph of the main transfer learning CNN model which represents how the model was trained and

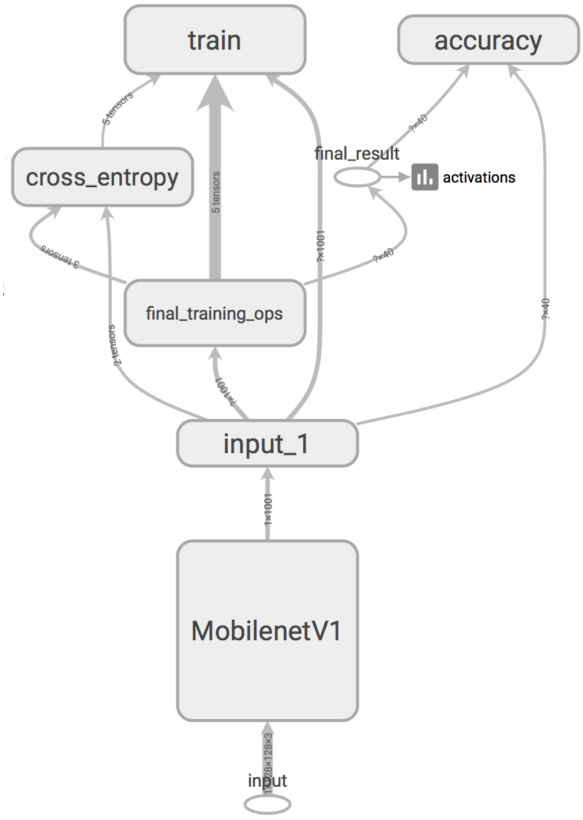

Fig. 6. Computational graph of the Transfer Learning Mobilenet model

the also gives a brief outline of the Transfer Learning CNN model flow.

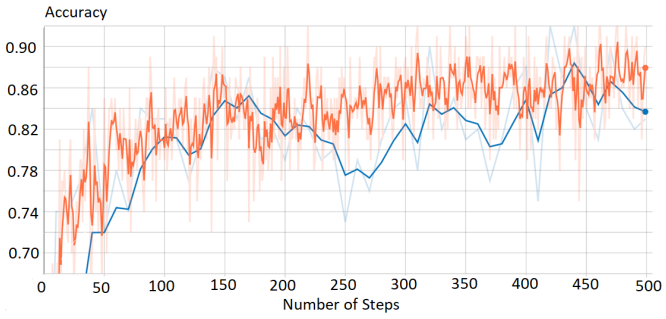

Fig. 7. Accuracy of training and validation data

Based on the above set parameters our main Transfer Learning model was able to achieve 88% accuracy on training data and 83.89% accuracy on validation data.

Error rate reduced to 45.80% on the training data, whereas on validation data it reduced to 48.77%.

We experimented by implementing various parameters and trained our model on different combinations of parameters to see which possible combination achieved the best result. We also trained our model on different learning rate [31] values and performed grid search to identify which value achieved higher accuracy.

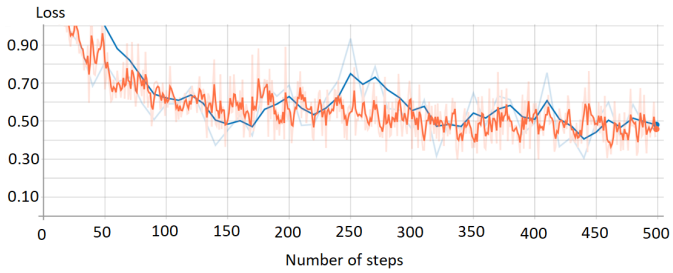

Fig. 8. Error rate of training and testing data

learning rate	train accuracy	validation accuracy	test accuracy
1.0	74.5%	71.94%	70.0%
0.5	76.81%	72.38%	71.83%
0.1	80.0%	79.32%	77.61%
0.05	86.41%	81.12%	80.48%
0.01	88.0%	83.89%	83.09%
0.001	95.19%	81.57%	79.3%

TABLE III
LEARNING RATE VARIATION

We found that as the learning rate decreases, the accuracy increases. However, if we keep too low a learning rate the accuracy decreases again. Our model was able to achieve the optimal accuracy when we set the learning rate to be 0.01. Table III shows the results of our experiment.

The goal of our experiment was to find the optimal point at which our model converges to something useful while keeping the learning rate high enough so that we don't have to spend too much time training it. Setting too low a learning rate can not only increase the training time but also cause overfitting [32]. We observe this when we set our learning rate to be 0.001. Though the models training accuracy is high, its validation and test accuracy are low. This is because of overfitting. Similarly, setting too high a learning rate can cause underfitting. We observe this when we set the learning rate to be 1.0.

Figure 9 shows the output yielded by our model. We show six images belonging to different classes and the probabilities that our system assigned to them. The facial recognition model was also trained using the similar configurations as the main RIPA CNN model. We achieved best accuracy when we set the learning rate to be 0.1. The facial recognition model was able to achieve 91.2% accuracy on training dataset and 87% accuracy on testing dataset. Figure 11 shows the output that our Facial Recognition CNN model was able to achieve. We show six images belonging to 6 different identities and the accuracy our model was able to achieve identifying them.

2) Effectiveness of the RIPA Mobile App: We have developed a mobile app for the RIPA system. In this round of experiments, we aim to evaluate its effectiveness. Our evaluation was based on the knowledge learned from 12,182 test dataset images. These images are of special importance as they have metadata like geotags, timestamps etc., associated

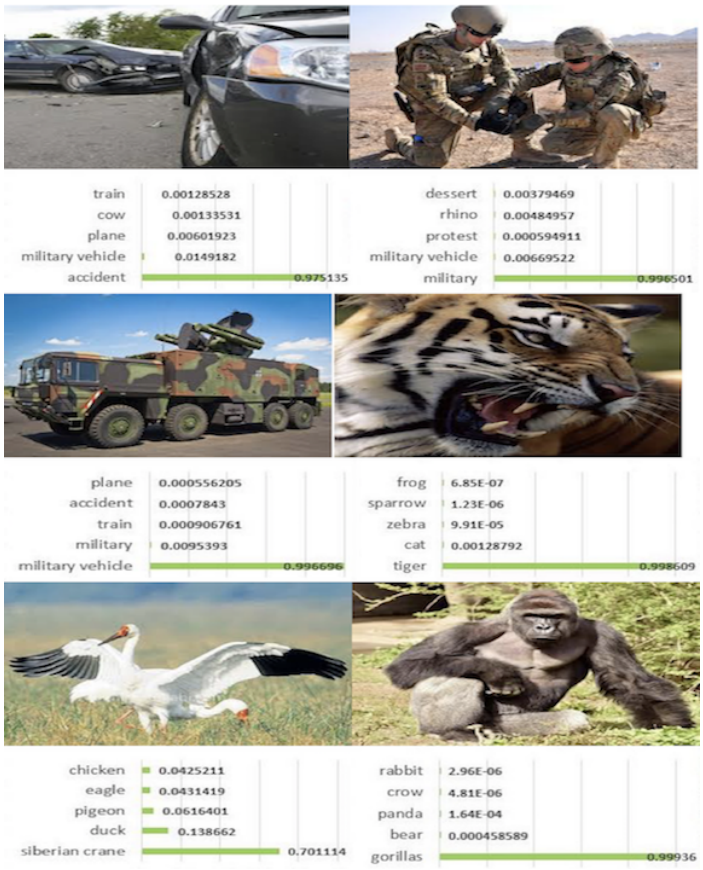

Fig. 9. Main Transfer Learning CNN evaluation

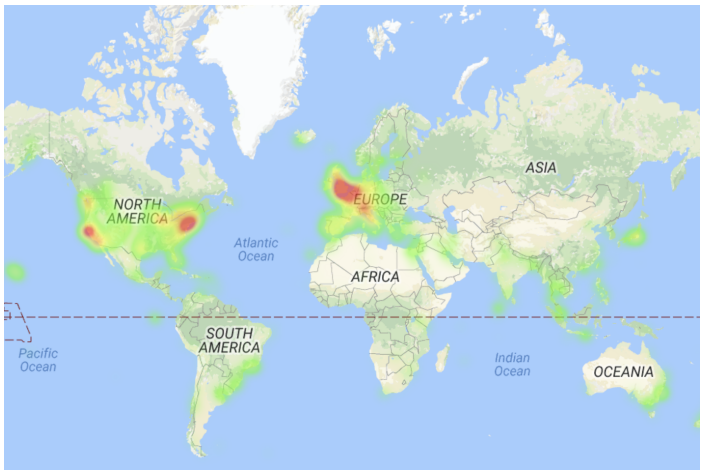

Fig. 10. Heat map of testing data images with geolocation metadata

with them.

Figure 10 visualizes the location of each image in the test dataset as a heat map. These images will enable the user to test and experience the functionalities of the application to full extent.

We tested the main interface by creating 5 custom datasets of 100 images each, of which 50 images belonged to non-sensitive and 50 images belonged to sensitive categories. We

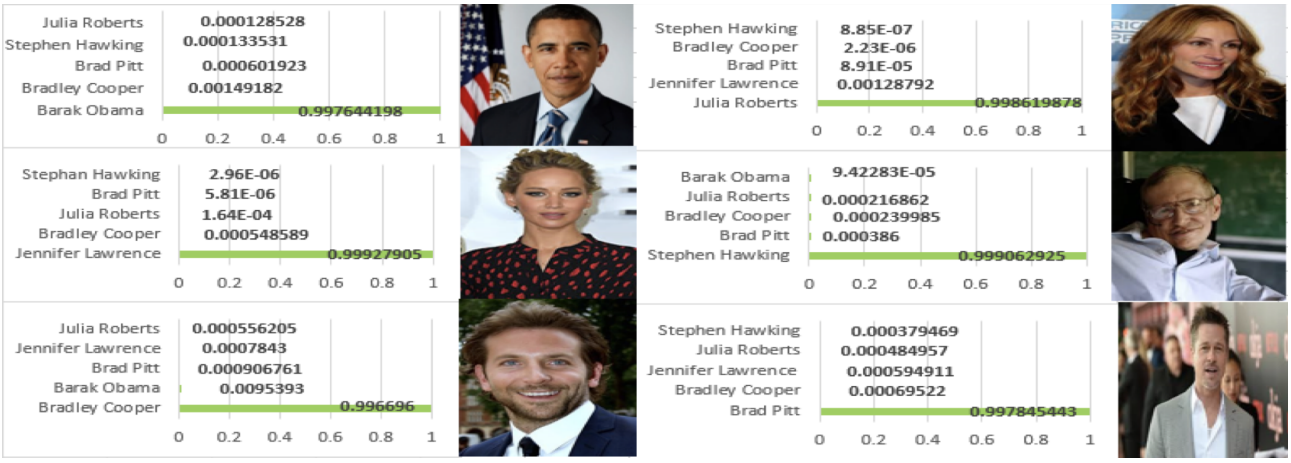

Fig. 11. Face recognition CNN evaluation

randomly selected 25 images from the custom dataset and upload them using the RIPA application to test how the system handled different inputs. Figure 12 shows the results generated by the RIPA system when an image containing an endangered species with group of people in the background was uploaded.

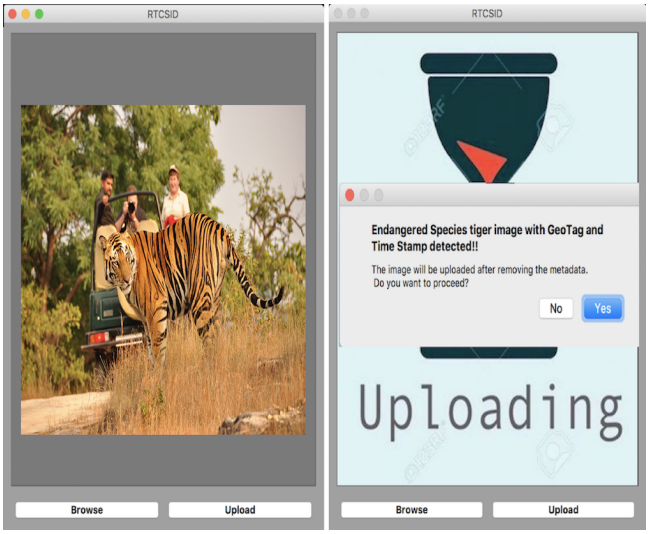

Fig. 12. RIPA interface

To test the facial recognition interface, we uploaded images of groups of people which included people whose facial features the RIPA facial recognition CNN model was trained to identify. The RIPA Facial Recognition system can recognize multiple individual faces within an image combining the usage of HOG methodology and convolution neural network. Using the predicted results from the Facial Recognition CNN as inputs, it quickly cross references its results with the RIPA dataset to check for any privacy and security conflicts, if any security flags are raised it will then issue privacy and security policies for the flagged identities. Figure 13 shows the results of the RIPA facial recognition interface.

Fig. 13. Facial Recognition interface

V. Conclusion and Future Work

In this paper, we have proposed a Real Time Image Privacy Alert (RIPA) System which helps classify and detect privacy-sensitive objects within the images being uploaded, and issue privacy alert to prevent the privacy violation. The proposed RIPA system is unique in that it not only considers the privacy of the photo owner, but also takes care of privacy preferences of other human or sensitive objects within the same photo which may be even in the background of the photo. The RIPA system combines advantages of deep learning and metadata analysis techniques to achieve the goal. A mobile application of the RIPA system is developed and evaluated in a large set of images.

As future work, we plan to build a multi-label image classification system using attention based neural network [33] that can help analyze the context of an image and gain more detailed insights about the relationships between multiple objects within the image.

ACKNOWLEDGMENT

This work was funded by National Science Foundation under project CNS-1651455 and CNS-1564101.

REFERENCES

[1] A. C. Squicciarini, S. Sundareswaran, D. Lin, and J. Wede, "A3p: adaptive policy prediction for shared images over popular content sharing sites," in *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*. ACM, 2011, pp. 261–270.

[2] J. Yu, B. Zhang, Z. Kuang, D. Lin, and J. Fan, "iprivacy: image privacy protection by identifying sensitive objects via deep multi-task learning," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1005–1016, 2017.

[3] S. Ahern, D. Eckles, N. S. Good, S. King, M. Naaman, and R. Nair, "Over-exposed?: privacy patterns and considerations in online and mobile photo sharing," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2007, pp. 357–366.

[4] C. E. Jacobs, A. Finkelstein, and D. H. Salesin, "Fast multiresolution image querying," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 1995, pp. 277–286.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[6] M. Ames and M. Naaman, "Why we tag: motivations for annotation in mobile and online media," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2007, pp. 971–980.

[7] A. D. Miller and W. K. Edwards, "Give and take: a study of consumer photo-sharing culture and practice," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. Acm, 2007, pp. 347–356.

[8] A. Besmer and H. Lipford, "Tagged photos: concerns, perceptions, and protections," in *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2009, pp. 4585–4590.

[9] J. Bonneau, J. Anderson, and G. Danezis, "Prying data out of a social network," in *Social Network Analysis and Mining, 2009. ASONAM'09. International Conference on Advances in*. IEEE, 2009, pp. 249–254.

[10] L. Fang and K. LeFevre, "Privacy wizards for social networking sites," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 351–360.

[11] H. R. Lipford, A. Besmer, and J. Watson, "Understanding privacy settings in facebook with an audience view." *UPSEC*, vol. 8, pp. 1–8, 2008.

[12] K. Liu and E. Terzi, "A framework for computing the privacy scores of users in online social networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 5, no. 1, p. 6, 2010.

[13] E. M. Maximilien, T. Grandison, T. Sun, D. Richardson, S. Guo, and K. Liu, "Privacy-as-a-service: Models, algorithms, and results on the facebook platform," in *Proceedings of Web*, vol. 2, 2009.

[14] R. Ravichandran, M. Benisch, P. G. Kelley, and N. M. Sadeh, "Capturing social networking privacy preferences," in *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, 2009, pp. 1–18.

[15] M. De Choudhury, H. Sundaram, Y.-R. Lin, A. John, and D. D. Seligmann, "Connecting content to community in social media via image content, user tags and user communication," in *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*. IEEE, 2009, pp. 1238–1241.

[16] H.-M. Chen, M.-H. Chang, P.-C. Chang, M.-C. Tien, W. H. Hsu, and J.-L. Wu, "Sheepdog: group and tag recommendation for flickr photos by automatic search-based learning," in *Proceedings of the 16th ACM international conference on Multimedia*. ACM, 2008, pp. 737–740.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[18] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *International journal of computer vision*, vol. 77, no. 1, pp. 157–173, 2008.

[19] T. Hastie, R. Tibshirani, and J. Friedman, "Overview of supervised learning," in *The elements of statistical learning*. Springer, 2009, pp. 9–41.

[20] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," *CoRR*, vol. abs/1502.00873, 2015. [Online]. Available: http://arxiv.org/abs/1502.00873

[21] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[22] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[23] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[24] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1891–1898.

[25] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724.

[26] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[27] B. Graham, "Fractional max-pooling," *arXiv preprint arXiv:1412.6071*, 2014.

[28] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[30] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[31] D. R. Wilson and T. R. Martinez, "The need for small learning rates on large problems," in *Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on*, vol. 1. IEEE, 2001, pp. 115–119.

[32] T. Dietterich, "Overfitting and undercomputing in machine learning," *ACM computing surveys (CSUR)*, vol. 27, no. 3, pp. 326–327, 1995.

[33] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.