Urban Tribe

Anonymous WACV submission

Paper ID ****

Abstract

We explore the use of pre-trained convolutional neural network features for social group recognition. A social group recognition framework is proposed. It takes in both individual and global features of group images, and shows promising results for group recognition task which significantly outperform previous results. We verify the generalization ability and semantic information of pre-trained CNN features. We show the necessity of fine-tuning in social group recognition, and this suggests a potential usage of pre-trained CNN features with adaptation for novel computer vision tasks.

1. Introduction

In the past few years, there have been impressive progress in understanding semantic meaning of images, such as object recognition, scene recognition, and object detection. The power of Convolutional Neural Networks (CNNs) has is especially noticeable. However, studies in analysis of a group of people in images are still deficient. Current search algorithms fail to capture information of personal styles or social characteristics of groups of people, but retrieve images with similar global appearance [8]. The analysis of groups of people is difficult in that the group categories are semantically ambiguous, and have high intra-class variance.

Recognition of groups of people from a social perspective provides many potential applications. With more accurate group searching results, more accurate recommendations can be made in social networks, and more relevant advertisement for particular groups of people benefits both consumers and sellers.

[12] and [8] studied this problem of group recognition. They created an urban tribe dataset consisting of 11 categories, with about 100 labeled per class. They proposed a group recognition pipeline. Rather than classifying isolated individuals in the group images, they focused on group features and models.

CNN architecture has been proved to achieve outstanding results in various computer vision tasks, and it's argued that deep architectures in CNN can capture visual features of different semantic level in the hidden units ([**?**]). Recently, features learnt for large scale recognition tasks with large amounts of training data have been used for new tasks, and the features outperform many conventional features in many tasks([**?**], [2]). This verifies that generic visual features are obtained from pre-trained CNN model.

this is what to modify later with results The most frequently used pre-trained model was proposed in [**?**] trained with ImageNet[1]. And the new tasks for which CNN features are used range from object recognition, scene recognition, to subcategory recognition. However, the objects or categories of these tasks are mainly covered in the 1000 ImageNet categories, and it is predictable that some features pre-trained on ImageNet are useful for the new tasks. Meanwhile, the social groups of individuals require semantic features of individual's style, which have little overlap with the 1000 ImageNet categories.

In this paper, we investigate the generalization ability of pre-trained CNN features to social group recognition. We propose a CNN feature based architecture for social group recognition. Our model takes in both individual feature and global scene feature finetuned from CNN pre-trained weights. Our result shows a boost of performance from the previous group classification method provided by [8]. We show that both individual information and global scene information contribute to a social group's characteristics, and that different feature extraction schemes for individual and global information is necessary. We also show the role of adapting generic pre-trained CNN features to social group images with finetune.

Further, we visualize the CNN features in different layers, and show a better semantic clustering with ascending layer, validating the feature composition of our framework. Finally, we analyze the properties of the social groups and the urban tribe dataset, showing that the representation our model achieves is in a high semantic level.

[1] http://image-net.org/challenges/LSVRC/2012/browse-synsets

2. Related Work

Convolutional Neural Networks(CNNs) with back-propogation were introduced around 1990's by [10]. Since then, CNNs showed successful results on various computer vision tasks, such as hand-written digit classification ([11]), classification task on ImageNet dataset ([**?**]).

Recently, many researches show the ability of generalization of pre-trained CNN features on large dataset such as ImageNet. [**?**] shows excellent generalization of the pre-trained CNN features. They kept all the layers of ImageNet-trained model fixed except last softmax classifier, and achieved bets results on Caltech-101 and Caltech-256. [2] used different layers of pre-trained CNN network as features and trained simple classifiers such as SVM and Logistic Regression, and outperformed the state-of the-art on several vision challenges such as scene recognition and domain adaptation. GoogLeNet [2] used improved CNN architecture pre-trained on ImageNet dataset, and won the the detection task of Large Scale Visual Recognition Challenge 2014 (ILSVRC2014).

[8] created an urban tribe dataset consisting of 11 classes. The classes are defined from social group labels provided by Wikipedia. They selected the eight most popular categories from their list of subculture, and added three other classes corresponding to typical social venues in addition. For each classes, images of groups of people are searched with different search engines, and a broad range of scenarios for each class are collected. [12] and [8] also provide a group description and several classification methods. Group description consists of person descriptors and global group descriptors: Six part of person is detected, and a set of predefined descriptors are computed for each part; Global descriptors use a both low level and high level descriptors to describe the context and group properties of the image. Two options of classification methods are provided: bag of parts-based classification and SVM-based classification.

Categorizing the social groups of individuals belongs to fine-grained classification task which recently draws more interest of the computer vision community. It aims at giving the fine-grained categories in a certain class. Fine-grained classification is more difficult than conventional classification tasks, because the categories are semantically as well as visually similar, and are even challenging for humans. The Fine-grained Challenge 2013 (FGComp) provided the data in several categories including aircraft, birds, cars, dogs, and shoes. [5] achieved the best result using classifier based on fisher vectors. However, CNN based methods using [6] or [2] gave not so good results, especially when the bounding box of test data is missing.

There is some research in analyzing social groups of people. [4] showed the visual structure of a group helps understanding events. [13] showed social relationships modeling helps people recognition. [1] used both local and global factors for group level expression analysis.

[2]http://www.image-net.org/challenges/LSVRC/2014/results

3. Methods

This section describes the urban tribe dataset and elaborates on the model architecture.

3.1. Urban tribes dataset

Urban tribes are groups of people who have similar visual appearances, personal style and ideals. The urban tribes dataset consists of 11 different categories: biker, country, goth, heavy-metal, hip-hop, hipster, raver, surfer, club, formal, casual/pub, with an average of 105 images from each category.

Different from conventional visual classification problem, urban tribe categories are more ambiguous and subjective. Also, each class contains a broad range of scenarios. The high intra-class variation of the urban tribe dataset makes the classification task challenging. The urban tribe dataset also has some interesting properties. The number of people in each urban tribe image varies. Members in one tribe often have similar visual styles, including their clothes, accessories, and even demeanor. For example, surfers possibly carry surfboards, and the goth often have dark attire, makeup and hair. The environment they are in also contributes to each tribe characteristics: pictures of countries are more likely to be taken outdoor with grassland, while pictures of clubbers are often photographed in clubs with dim lightings.

3.2. Classification hierarchy

To utilize the properties of urban tribes fully, our feature vector consists of both elements: individual features and environment features. For each feature type, we use similar extraction strategy. Individual features and environment features are hierarchically combined to form the final decision function. The network hierarchy is shown in Figure 1.

For each group image, we represent the group G as the combination of a set of people and the environment. To give the prediction of class C, the individual feature vectors and environment feature vectors are extracted separately.

For the individual feature vectors, first, individual person candidates are detected with a poselet based person detection algorithm. The person candidate images $H = \{H_1, H_2, ..., H_p\}$ are used as a whole instead of a set of body part bounding boxes. Each person candidate is resized to 256×256, and ten 227×227 patches $\{h_{ij}\}, i \in \{1, 2, .., p\}, j = 1, 2, ..., 10$ are extracted using the same method as fine-tune image set generation.

Each Individual image patch h_{ij}then passes through the Convolutional Neural Network for person images

CNN_{Person}, generating activations from the 6th and 7th hidden layer. The activations from 6th and 7th layer are both in 4096 dimensions. They are concatenated to form a 8192-dimensional vector f_{ij}, where $i \in \{1, 2, .., p\}, j \in \{1, 2, ..., 10\}$.

The feature vectors are then fed into a multi-class SVM_{Person}. We use LIBLINEAR[3] to train the SVM on individual patches, and to estimate probabilities for each category given individual patch h_{ij}: $\Pr_{ij}(C|h_{ij}), C \in \{1, 2, .., c\}$, where c is the number of classes in urban tribe dataset. The individual patches h_{ij} in one group image are usually highly correlated. Therefore, in order to obtain a reliable probability estimate from the noisy yet correlated set of probabilities $\Pr_{ij}$, a simple but effective average pooling is performed to $\Pr_{ij}$:

$$\Pr_{People}(C|H_1, ..., H_p) = \frac{1}{10p} \sum_{i,j} \Pr_{i,j}(C|h_{ij}) \quad (1)$$

$\Pr_{People}(C|H_1, ..., H_p)$ is the probability estimate of class C given the set of people candidate images H.

On the other hand, the entire environment in the scene image, denoted by S, is directly utilized for probability estimation. The procedure to generate probability estimate of class C given the environment as a whole $\Pr_{Scene}(C|S)$ is similar with that for $\Pr_{People}(C|H)$. The difference is that the input of Convolutional Network is 227×227 patches extracted from the entire scene image, and the fine-tuned Convolutional network: CNN_{Scene} and SVM: SVM_{Scene} are trained with the training set of entire scene images. Several different strategies to extract patches from scene images and corresponding Convolutional Neural Network architectures are explained in Section 3.3.

Therefore, the probability estimate of a class C given observation of scene S is:

$$\Pr_{Scene}(C|S) = \frac{1}{K} \sum_{k=1}^{K} \Pr_{k}(C|s_k) \quad (2)$$

where K is the number of scene patches extracted from one group image, and this number varies with different patch extraction strategies. s_k is the kth scene patches. $\Pr_k(C|s_k)$ is the probability for class C given kth scene patch. Average pooling is still used here, because the assumption of high correlation in patches holds.

Now we have the estimates of two kinds of conditional probability $\Pr_{People}(C|H)$ and $\Pr_{Scene}(C|S)$. We make a strong assumption that the two types of features are independent, and that the prior probability distribution of the urban tribes $\Pr(C)$ is a uniform distribution. The classification problem can be expressed as maximizing the objective function:

$$L = \arg\max_{i=1,...,c} \Pr(C = i|G) \quad (3)$$

where

$$\begin{aligned} \Pr(C = i|G) &= \Pr(C = i|H, S) \\ &= \frac{\Pr_{People}(C = i|H_1, ..., H_p) \cdot \Pr_{Scene}(C = i|S)}{\Pr(C = i)} \\ &\propto \Pr_{People}(C = i|H_1, ..., H_p) \cdot \Pr_{Scene}(C = i|S) \end{aligned} \quad (4)$$

and L is the predicted label for the group image.

3.3. Convolutional network feature extraction

It is shown in many experiments that a set of weights of convolutional network trained from ImageNet can generate a set of generic visual features.

Following [6]'s work, we use the network framework called Caffe. The network architecture is described in [**?**], which won the ImageNet Large Scale Visual Recognition Challenge 2012. We take the activations from the 6th and 7th hidden layer of the convolutional neural network, which are two fully connected layers before the class prediction layer. We also take the activations from 6th or 7th layer alone as comparison. We choose these two layers, because as the layers ascend, the features extracted show increasing invariance and semantic meaning.

We use the pre-trained set of weights of the network released by Caffe as the initial parameters of our network. The pre-trained model was trained on Imagenet ILSVRC-2012, and all images are first resized to 256×256 before they can be used as inputs to the network.

3.3.1 Pre-processing of the dataset

The urban tribe dataset is a relatively small dataset, and both people candidate crops and scene images are of various resolution. Our convolutional neural networks requires constant input image size of 227×227, so pre-processing of the dataset is necessary.

There are several strategies to make one image compatible with the CNN:

1. As in [**?**], resize the image to a fixed resolution of 256×256, and crop five 227×227 patches (from four corner and the center) and their horizontal reflections to generate ten patches from one single image. This way, the aspect ratio of the original images are lost, but for each crop, the portion it takes from the original image is fixed, so that the amount of information all the crops have is relatively stable.

2. Keep the aspect ratio of the original image, resize the shorter side to 256, and then crop five 227×227 patches and their horizontal reflections as mentioned. This method avoids distortion of the image and objects in it, but the crops will possibly lose much information

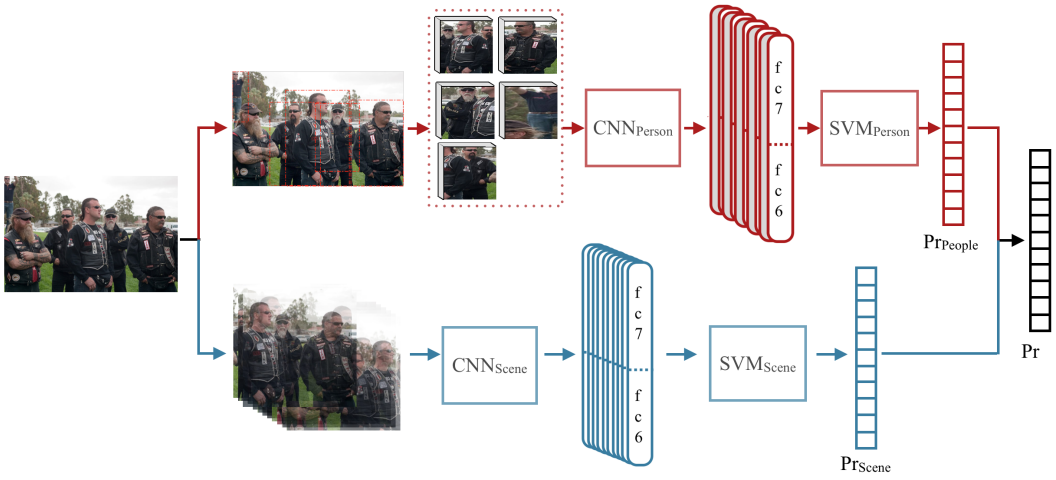


Figure 1: Architecture of classification algorithm using Nets$_{SDense}$. The upper half estimates the probability given people candidate images, and the lower half estimates the probability given the entire scene. Dense crop CNN$_{Scene}$ and distorted crop CNN$_{Person}$ are used.

when the aspect ratio of the original image is far away from 1.

3. Keep the aspect ratio of the original image, resize the shorter side to 256, and then densely crop multiple 227×227 patches and their horizontal reflections. This way, the information of original image is kept by dense cropping process, and the distortion is avoided. The number of crops attained with this method is larger than the previous two methods.

3.3.2 Network Finetune

Although the pre-trained network from [6] can already generalize well to many datasets, the urban tribe dataset has its unique property. It emphasizes certain visual features such as clothes, while pays less attention to other visual features. Also, it emphasizes style of the object rather than distinct category. To rearrange the importance of different features and adjust the features to adapt to urban tribe dataset, the network can be finetuned.

The dataset used for finetune is the same set used for future training. The input patches of the Convolutional network is of size 227×227. The initial Convolutional network has 1000 outputs in final layer, corresponding to 1000 class-wise probability predictions. In our finetune process, the last layer is replaced by 11 probability prediction outputs, and the initial weights of the last layer connection are initialized to have zero mean gaussian distribution. Back propagation is used, and the learning rate is set to be small so that the fine-tune process adapts the extracted features to urban tribe dataset while preserving the initial property in general: the initial learning rate used for pretraining is 0.01; We set the initial finetune learning rate of the parameters except for the last layer as 0.001, and keep the initial learning rate for last layer 0.01, because the last layer is not pre-trained.

3.3.3 Network modification

The side effect of dense cropping technique in Section 3.3.1 is that there is a boost in the number of input crops, and the speed of feature extraction is largely slowed down.

We apply a trick to speed up the process. In a convolutional neural network, a fully connected layer can be viewed as a convolutional layer with 1×1 sized kernels. Therefore, in our CNN architecture, we can substitute convolutional layers for fully connected layers of 6th and 7th layer. For example, the input of 7th layer is 4096-dimensional feature, and the output is 4096-dimensional feature. The layer has 4096×4096 connections. We can view the connections as 4096×4096 convolutional filters with size 1×1. For one input image of size 227×227, the output of the 7th layer is 4096 feature maps of size 1×1, while for one input image of arbitrary size (larger than 227×227), the output of the 7th convolutional layer is 4096 feature maps of larger size, each 4096-dimensional element corresponds to one 227×227 cropping of the input image. Now the modified network can take images with arbitrary size (larger than 227×227) as input, and extract patch features much more efficiently.

3.3.4 Choices of network combination

Scene images and individual images have different properties, and need different strategies of pre-processing and separate fine-tuning. For scene network CNN_{Scene} and scene images, due to the small size of the dataset, we use the dense cropping technique to increase the dataset. Only the center crops of each original group image and their mirrors are used for finetune. For person network CNN_{Person} and corresponding input, we use the first cropping technique, because the subimages have normally long height and short width, and the second and third strategies using squared crops of a person image will lose much information, no matter which location we choose to crop them; whereas the first method ensure each crop keeps the essential features for classification. Finetuning of CNN_{Person} uses center crops and their mirrors of person candidate images.

The combination of dense crop CNN_{Scene} and distorted crop CNN_{Person} are denoted as Nets_{SDense}.

We also construct other combination of networks for comparison:

1. Nets_{NoTune}: Directly use the pre-trained network by [6] for both scenes and persons, and use the first cropping technique (distorted crops) as input patches for both networks. This choice of cropping strategy is in consistent with the way the network is pre-trained.

2. $\text{Nets}_{SSparse}$: Use the second cropping strategy for scene features, and the first cropping strategy for person features. Finetune procedure is the same as Nets_{SDense}.

3. $\text{Nets}_{SDistort}$: Use the first cropping strategy for both scene features and person features. Training set for finetune procedure is the center crops and their mirrors of scene/person images.

4. Experiments and Results

In this section, the performance of the proposed classification scheme is evaluated and analyzed. In the experiments, six rounds of 5-fold cross validation are performed, therefore we have 30 training experiments in total. Dataset is partitioned into 5 equal sized subsets, containing one-fifth of the data points from each category. One of the subsets is used as test set, and the remained 4 subsets are used as training data. In the finetune procedure, there are 6000 training iterations, and learning rate is decreased by ten times after each 1000 iterations.

4.1. Urban tribe classification performance

Table 1 shows the comparison of performance using different approaches. The 30 segmentations of datasets are used for all the approaches tested in this section, and 30 test results are averaged for each approach. The standard error is shown with the accuracy in Table 1. We also compare our result with the result achieved by [8] using their best model. The advantage of CNN pre-trained features is obvious.

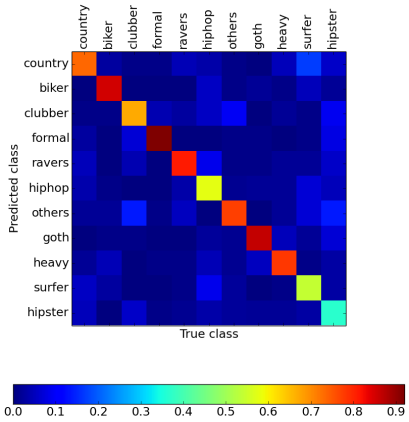


Figure 2: Confusion matrix for classification results with Nets_{SDense}, using people and scene features.

The confusion matrix is shown in Figure 2 for Nets_{SDense}, and all the 30 training experiments are averaged. We can observe there is a obvious difference of difficulty of different categories. Class *formal* has accuracy as high as about 90%, while class hipster is the most difficult class, having less than 60% accuracy.

Comparing the result of using different features in the same approach shows the necessity of every step of our architecture. In results using Nets_{SDense} with concatenated features, average accuracy for each individual person candidate is low as 47.10%. Average pooling of individual candidate probability estimates produces a large accuracy increase of about 20%. Accuracy using the entire scene only results in 67.26% accuracy. Combining probabilities $\text{Pr}_{People}(C|H_1, ..., H_p)$ and $\text{Pr}_{Scene}(C|S)$ achieves accuracy as high as 71.22%, which verifies the complementary role of people candidate feature and environment feature in a group image.

We also compare vertically the accuracy of different approaches, to show the role of network feature concatenation. Using only 7th layer or 6th layer activation from the networks Nets_{SDense} produces decent results, showing both layers' activations can generate high semantic features. Concatenating both layers' activations increases the accuracy by 0.5%, indicating the slight information loss of the 7th fully connected layer.

To see the role of finetuning, we can compare the result of Nets_{NoTune} and $\text{Nets}_{SDistort}$. These two approaches

Table 1: Performance of different approaches using different information.

Accuracy (%)	Individual candidate	People	Entire scene	People+Scene
Nets$_{SDistort}$ with concatenated features	39.99 ± 0.30	64.09 ± 0.63	62.77 ± 0.52	69.28 ± 0.50
Nets$_{SDense}$ with fc7 features	47.07 ± 0.30	67.09 ± 0.52	65.05 ± 0.42	70.74 ± 0.47
Nets$_{SDense}$ with fc6 features	45.84 ± 0.34	66.20 ± 0.46	67.08 ± 0.51	70.43 ± 0.46
Nets$_{SDense}$ with concatenated features	$\mathbf{47.10 \pm 0.34}$	$\mathbf{67.29 \pm 0.50}$	$\mathbf{67.26 \pm 0.50}$	71.22 ± 0.46
Nets$_{SSparse}$ with concatenated features	$\mathbf{47.10 \pm 0.34}$	$\mathbf{67.29 \pm 0.50}$	66.81 ± 0.42	$\mathbf{71.23 \pm 0.49}$
Nets$_{SDistort}$ with concatenated features	$\mathbf{47.10 \pm 0.34}$	$\mathbf{67.29 \pm 0.50}$	65.35 ± 0.37	71.15 ± 0.50
SVM_8[8]	-	-	-	46(std: 2)

both use resizing that causes distortion, and they only vary in finetune procedure. There is a large performance improvement with finetune, both for person features and scene features. This shows the benefit of adapting existing generic Convolutional Network to specific dataset.

Nets$_{SDistort}$, Nets$_{SSparse}$, and Nets$_{SDense}$ use different patch extraction strategies. Note that we use the same distorted patch extraction method for person images, as mentioned in Section 3.3.4, while we use three different methods for scene images. The results for scene images show the advantage of keeping the aspect ratio of scene images, and the slight advantage of using dense crops. However, the final results with People+Scene for the three methods don't have significant differences, this is due to the combination with people information.

4.2. Convolutional Network feature analysis

Yufei: Is this section necessary? We use t-distributed stochastic neighbor embedding technique [9] to visualize the power of different layers' features in convolutional network. Two dimensional embedding of the high dimensional CNN feature space is extracted, and we plot the data as 2-d points with different colors indicating different classes they belong to. Powerful features will pull data of different semantic classes more apart.

We randomly choose one training-test partition and corresponding fine-tuned network parameters, and to avoid overfitting effect, we examine the test set of Nets$_{SDense}$ approach. In Figure 3, each data is plotted as a dot in each figure. The three columns correspond to data separation in first layer, fourth layer and seventh layer respectively. The first row visualizes features of all test data in scene CNN CNN$_{Scene}$, and second row picks three classes from the first row. The third row is features of CNN$_{Person}$, and the three classes are picked for the last row. The three classes chosen in second and last row is (goth:red, heavy:green, surfer:blue).

In both CNN$_{Scene}$ and CNN$_{Person}$, there is a clear trend of class separation. As the layer ascends, , the data from same class are more concentrate, and inter-class distance are larger. The selected three classes show the trend more clearly: they essentially form three clusters in the seventh layer. As shown in the second and last rows, person data is more challenging: The semantic classes are not as separate as with the scene features.

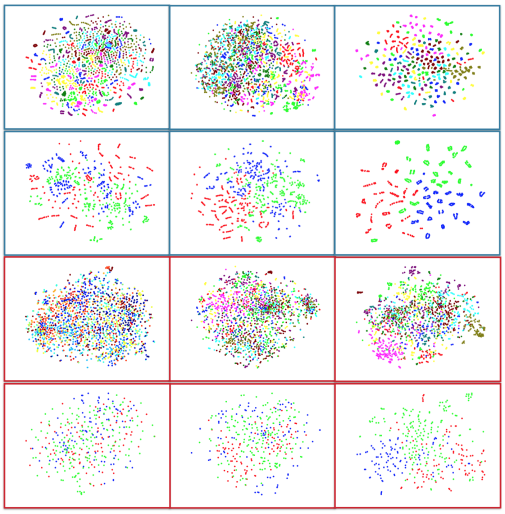


Figure 3: Feature visualization of Nets$_{SDense}$: CNN$_{Scene}$ (first two rows) and CNN$_{Person}$(second two rows). Rows: first: all scene images; second: all data from goth/heavy/surfer urban tribes; third: all test person images; fourth: all test data from goth/heavy/surfer urban tribes. Columns: first: first convolutional layer; second: fourth convolutional layer; third: seventh fully convolutional layer

4.3. Urban tribe classes vs. Imagenet classes

It's recently being acknowledged that CNN pre-trained features are generic and can be used for new tasks. In this section we check the relationship between the new tasks and original Imagenet task, using the urban tribe dataset,which gives us some insight about the features extracted from pre-trained network.

The urban tribe dataset contains groups of people, and the important features for categorization are mainly human related features, such as attire, make up, posture and expres-

sions. However, the ILSVRC dataset used for pretraining contains few images of humans. Instead of examining the output of the layers directly, we try to find the relationship between the 1000 classes in ILSVRC dataset and the classes in two image sets: urban tribe dataset, and person candidate images extracted from urban tribe dataset.

We use the parameters of pre-trained CNN network as our feature extraction model, and train a softmax layer on top of its 7th layer to predict the probabilities of one input image(either scene image or person candidate image) being in certain urban-tribe class $\Pr(l_{urban})$, where l_{urban}pre-trained is the 11 urban tribe categories. The output layer is trained for 3000 training iterations. We can also use the output of the pre-trained CNN network to predict probabilities of one input image being in certain Imagenet category, denoted as $\Pr(l_{Imagenet})$, where $l_{Imagenet}$ is the 1000 Imagenet categories. We use one round of 5-fold cross validation, and use all the images in urban tribe dataset for analysis.

We first check the relationshape of $\Pr(l_{urban})$ and $\Pr(l_{Imagenet})$ of person candidate images. We calculate the correlation coefficient of $\Pr(l_{urban})$ and $\Pr(l_{Imagenet})$ for all 1000 Imagenet classes, denoted as $R(\Pr(l_{urban}), \Pr(l_{Imagenet}))$. We also calculate the 1000 mutual information of the predicted score of urban tribe and Imagenet (where predicted score is 1 if the predicted label is the category being tested, 0 otherwise), denoted as $I(l_{urban}, l_{Imagenet})$.

In Figure 4, we choose two urban classes: *biker* and *hipster*, and plot the correlation coefficient R and mutual information I.The first row shows the result of *biker*, and second row *hipster*. For *biker*, there are several impulses in correlation plot, and one significant impulse in mutual information plot. *whiptail lizard* has high correlation with *biker*. Meanwhile, for *hipster*, which is the most difficult class, the correlation coefficient and mutual information are both low for all Imagenet classes.

To confirm the correlation,we use $\Pr(l_{Imagenet})$ directly as features, substituting the concatenated fc7fc6 features, and use the $\text{Nets}_{SDistort}$ approach for classification. The accuracy is 52.68%. This decent result indicates the relationship between l_{urban} and $l_{Imagenet}$. Then, we check $l_{Imagenet}$ with highest correlation coefficients with l_{urban}. In Figure 5, we choose four l_{urban}: *formal*, *ravers*, *goth*, *hipster*, and choose some examples of person candidate images that have both high $\Pr(l_{Imagenet})$ and high $\Pr(l_{urban})$. We also show examples of the images in $l_{Imagenet}$. We can see some of the shared features between corresponding person images and Imagenet images, for example, similar shape for women tops and stingrays (Figure 5b).

There is a correlation between class-wise accuracy of predicted l_{urban} and the degree of relationship between predicted $\Pr(l_{urban})$ and $\Pr(l_{Imagenet})$, as shown in Figure 6.

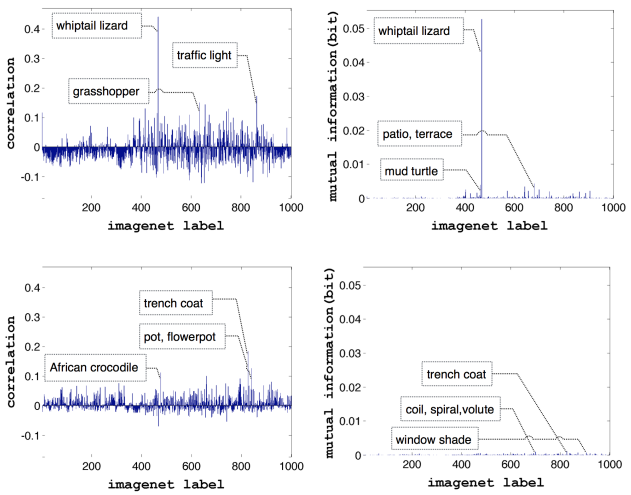


Figure 4: Correlation and mutual information of l_{urban} and 1000 $l_{Imagenet}$. The first row is for l_{urban}=Biker. The second row is for I_{urban}=Hipster. Top three Imagenet classes are marked.

Class-wise accuracy is calculated for person candidate images(Figure 6a, 6b) or scene images(Figure 6c, 6d). For each l_{urban}, the maximum correlation/mutual information over 1000 Imagenet classes are used to indicate the degree of its relationship with $l_{Imagenet}$.

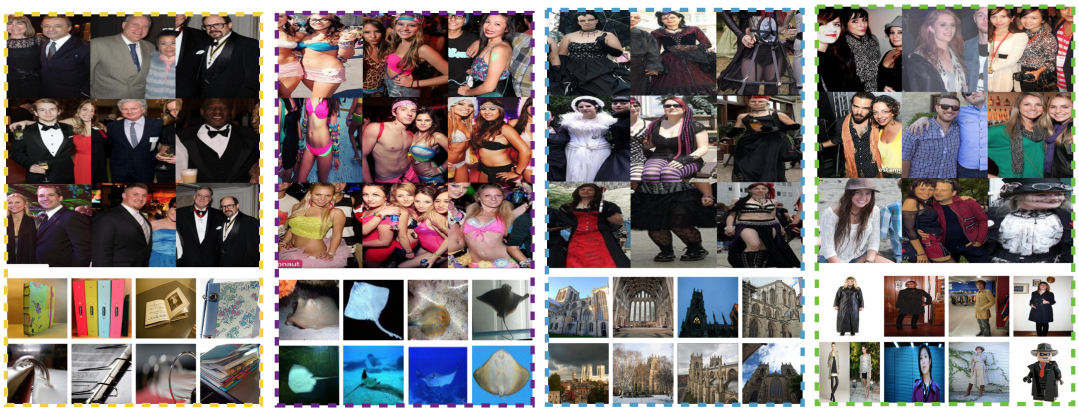
The correlation between Imagenet class and urban tribe class and its relationship with class-wise recognition rate may indicate that the "generic" features extracted by pre-trained CNN networks are not so generic. The network is trained to separate the Imagenet classes most, and if we use the features for a new classification task, the performance of the task is related to how well the new classes can be "mapped" to the Imagenet classes.

5. Conclusion

In this work, we explore the use of pre-trained deep network features for social group recognition. We propose a recognition framework which takes in both individual and global features. Our results show the success of our framework. Both visualization and numeric results show the generalization ability and semantic information of pre-trained CNN features. Our results also show the necessity of fine-tuning in social group recognition and imply the potential usage and modification of pre-trained CNN features for other computer vision tasks.

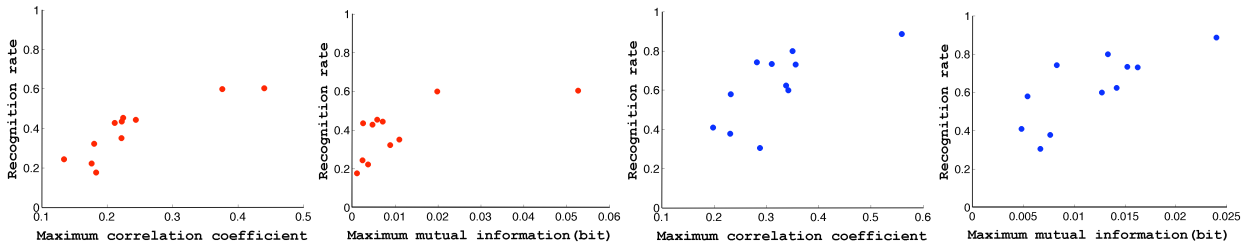
References

[1] A. Dhall, J. Joshi, I. Radwan, and R. Goecke. Finding happiest moments in a social context. In K. M. Lee, Y. Matsushita,



(a) Formal - Binder (b) Ravers - Stingray (c) Goth - Church (d) Hipster - Trench coat

Figure 5: selected urban tribe classes and the corresponding highest correlation $l_{Imagenet}$. Upper nine images: person candidate images with high $\Pr(l_{Imagenet})$ and high $\Pr(l_{urban})$. Lower eight images: example of images in $l_{Imagenet}$.



(a) Individual accuracy - maximum of R (b) Individual accuracy - maximum of I (c) Scene accuracy - maximum of R (d) Scene accuracy - maximum of I

Figure 6: The relationship between class-wise recognition rate and maximum of correlation $R(\Pr(l_{urban}), \Pr(l_{Imagenet}))$, class-wise recognition rate and maximum of mutual information $I(l_{urban}, l_{Imagenet})$

J. M. Rehg, and Z. Hu, editors, *ACCV 2*, volume 7725, pages 613–626. Springer, 2012.

[2] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013.

[3] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[4] A. C. Gallagher and T. Chen. Understanding images of groups of people. In *CVPR*, pages 256–263. IEEE, 2009.

[5] P.-H. Gosselin, N. Murray, H. Jégou, and F. Perronnin. Inria+xerox@fgcomp: Boosting the fisher vector for fine-grained classification. Technical Report 0, INRIA, December 2013.

[6] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/, 2013.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *NIPS*, pages 1106–1114, 2012.

[8] I. S. Kwak, A. C. Murillo, P. Belhumeur, S. Belongie, and D. Kriegman. From bikers to surfers: Visual recognition of urban tribes. In *British Machine Vision Conference (BMVC)*, Bristol, September 2013.

[9] G. H. Laurens van der Maaten. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, November 2008.

[10] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551, 1989.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324, 1998.

[12] A. C. Murillo, I. S. Kwak, L. Bourdev, D. Kriegman, and S. Belongie. Urban tribes: Analyzing group photos from a social perspective. In *CVPR Workshop on Socially Intelligent Surveillance and Monitoring (SISM)*, Providence, RI, June 2012.

[13] G. Wang, A. C. Gallagher, J. Luo, and D. A. Forsyth. Seeing people in social context: Recognizing people and social relationships. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *ECCV*, volume 6215, pages 169–182. Springer, 2010.

[14] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.