Efficient Gender Classification Using a Deep LDA-Pruned Net

Qing Tian, Tal Arbel, James J. Clark
Centre for Intelligent Machines & ECE Department, McGill University
845 Sherbrooke Street W, Montreal, QC H3A 0G4, Canada
{qtian,arbel,clark}@cim.mcgill.ca

Abstract

Many real-time tasks, such as human-computer interaction, require fast and efficient facial trait classification (e.g. gender recognition). Although deep nets have been very effective for a multitude of classification tasks, their high space and time demands make them impractical for personal computers and mobile devices without a powerful GPU. In this paper, we develop a 16-layer, yet lightweight, neural network which boosts efficiency while maintaining high accuracy. Our net is pruned from the VGG-16 model [39] starting from the last convolutional layer (Conv5_3) where we find neuron activations are highly uncorrelated given the gender. Through Fisher's Linear Discriminant Analysis (LDA) [8], we show that this high decorrelation makes it safe to discard directly Conv5_3 neurons with high within-class variance and low between-class variance. Using either Support Vector Machines (SVM) or Bayesian classification on top of the reduced CNN features, we are able to achieve an accuracy which is 2% higher than the original net on the challenging LFW dataset and obtain a comparable high accuracy of nearly 98% on the CelebA dataset for the task of gender recognition. In our experiments, high accuracies can be retained when only four neurons in Conv5_3 are preserved, which leads to a reduction of total network size by a factor of 70X with an 11 fold speedup for recognition. Comparisons with a state-of-the-art pruning method [12] in terms of convolutional layers pruning rate and accuracy loss are also provided.

1. Introduction

In recent years, deep learning has revolutionized many computer vision areas due to high accuracy for a wide variety of classification tasks. Although artificial neural networks have been used for visual recognition tasks since the early 1980s [20], recent algorithms have been success at training large networks efficiently [15, 6, 26, 16]. Given the huge amount of data that has become available, recent advances in computing have led to the emergence of deep neural nets. Even though deep learning techniques become the state-of-the-art solutions for various vision tasks, the requirement of a powerful GPU has made their wide deployment on general purpose PCs and mobile devices impractical. Moreover, from the 'very' deep VGG-Net [39] and GoogLeNet [41] to the 'extremely' deep Microsoft ResNet [14], the competition for higher accuracy with ever larger depths appears to be endless, rendering real-time performance on mobile devices even more out of reach.

In this paper, we explore ways to greatly prune very deep networks while maintaining or even improving on their classification accuracy. Our motivation stems from the current popular practice where, rather than train a deep net from scratch using all the available dataset, algorithm developers usually adopt a general network model and fine-tune it using a smaller dataset for the particular task. Therefore, there is a chance that some structures from the pre-trained model are not fully used for the current purpose. Our premise is that less useful structures (together with possible redundancies) could be pruned away in order to increase computational efficiency. Deep convolutional networks are generally considered to be composed of two components: the convolutional (conv) layers (alternated with activation and pooling layers) as feature extractors and fully connected (FC) layers as final classifiers [1]. Deep nets outperform many traditional computer vision algorithms mainly because the first component can learn very complicated features based on primitive ones given enough training data. More often than not, such features learned for a particular task are superior to handcrafted features designed with limited domain knowledge. The second component, FC layers, is essentially similar to logistic regression classifiers, which model the log-odds with a linear function. In this paper, we increase efficiency for each of the two components. We first investigate the firing patterns of last conv layer neurons through Fisher's Linear Discriminant Analysis (LDA) [8] and discover that those neuron activations are highly decorrelated for each class, which permits discarding

[1] In this paper, FC layer is used in a general sense and includes all the layers after Conv5_3.

a large number of less informative neuron dimensions without loss of information. As a result, the network complexity can be significantly reduced, which not only makes feature extraction more efficient, but also simplifies the classification task. In the second component, we analyze possible alternatives to the FC layers for the final classification. Instead of the FC layers, which model the log-odds based on linear functions, we explore multiple alternatives such as the Bayesian classifier and SVMs (with linear and RBF kernels). Although our approach is generally applicable to a wide range of attribute recognition problems, we use gender recognition as a specific example of facial attributes classification. Our experimental results show that when using the CNN features previously extracted, both a Bayesian and an SVM classifier with a RBF kernel can outperform using the original net with FC layers when the dataset is particularly challenging (e.g. partial occlusions, large view changes, complex backgrounds, blurs exist). Also, the combinations of LDA-Pruned CNN nets and the Bayesian/SVM classifiers take far less space (only a few megabytes) than the original net (over 500 MB) while having an 11 times faster recognition speed. In addition, we have analyzed the relationship of accuracy change and parameters pruned away, and have compared our approach to a state of the art pruning method in the literature [12]. According to the results, our Fisher LDA based pruning enjoys a lower accuracy loss, especially when the conv layers' pruning rate is high (say above 85%). Furthermore, unlike [12], our pruning approach can directly lead to space and time savings. The remainder of the paper is structured as follows: the relevant literature is reviewed in Section 2. In Section 3, our light weight deep networks along with alternative classifiers are introduced. Section 4 describes our experimental validation and compares our modified nets to their originals as well as other pruned structures in terms of accuracy and efficiency. In Section 5, our contribution and possible future directions are discussed. Section 6 concludes the paper.

2. Related Work

2.1. Gender Recognition

Traditional approaches for gender recognition are based on hand-engineered features that can be grouped to be either global [10, 27, 18, 2] or local [36, 5, 47, 44, 45]. Perez *et al*. [30] used a combination of both kinds of features with mutual information. O'Toole *et al*. [28] showed that depth information can also be helpful. Most of the above methods were tested on the highly controlled FERET benchmark [31] where near-perfect accuracies have been achieved [45, 30]. Shan [37] employed the Local Binary Patterns (LBP) and reported satisfactory results on some frontal or near frontal face images of the Labeled Faces in the Wild (LFW) dataset [17]. In terms of classification, SVMs are widely used alone [27] or with boosting algorithms such as Adaboost in [37] and can produce relatively high accuracies on simple datasets. When it comes to more challenging cases, such as when many occlusions and view changes are present, Toews and Arbel [44] demonstrated Bayesian classifiers' superiority to SVM when using a multiple local scale-invariant features based Bayesian framework. We refer our readers to [24, 34] for more traditional works. The main problem with handcrafted features based approaches is that they do not generalize well and require extensive domain knowledge, which is not always available.

The use of artificial feed-forward neural networks, for use in classification tasks, have been around for decades. In the 1990s, they began to be employed for gender classification [9, 32, 10]. However, the shallow structure of early neural networks has constrained their performance and applicability. It was not until late 2012 when Krizhevsky *et al*. [19] won the ImageNet Recognition Challenge with a ConvNet that neural networks regained attention. In the following years, various deep nets were successfully applied to a variety of visual recognition tasks including gender classification. Verma *et al*. [46] showed that the CNN filters correspond to similar features that neuroscientists identified as cues used by human beings to recognize gender. Inspired by the dropout technique in training deep nets, Eidinger *et al*. [7] trained a SVM with random dropout of some features and achieved promising results on their relatively small Adience dataset, on which Levi and Hassner [22] later trained and tested a not-very-deep CNN. Instead of training on entire images, Mansanet *et al*. [25] trained relatively shallow nets using local patches and reported better accuracies than whole image based nets of similar depths. To gain better accuracy, neural nets of larger depths are desired. However, the larger the depth is, the more parameters are needed to train, and the more data is required. To this end, recently relatively large-scale face datasets with gender labels were either created [35] or adapted. Liu *et al*. [23] provided 40 attributes labels (including gender) for the popular LFW [17] and CelebFaces [40] datasets, which opens up new opportunities for our research.

2.2. Deep Neural Networks Pruning

From the 8-layer AlexNet [19] to the ultra deep Microsoft ResNet [14], the trend of neural networks in the past five years can be summarized in one word: deeper. The struggle for higher accuracy with more layers reignited research in network pruning in order to reduce complexity. Earlier work, targeting shallow nets, include magnitude-based biased weight decay [33], Hessian based Optimal Brain Damage [21] and Optimal Brain Surgeon [13]. More recently, aiming at deep networks, Han *et al*. [12] developed a strategy to learn which connections are more important based on backpropagation. In [11], they added two

more stages of weight quantization and Huffman encoding in order to further reduce the network complexity. Their pruning is based on unit length connection, thus it may not well reflect larger scale utilities. Additionally, like other weight value based pruning methods, it assumes that large weight values represent high importance, which is not always the case (more explanations in Section 3.3). In terms of implementation, masks are required to disregard pruned weights during network operation, which inevitably adds to the computational and storage burden. To better utilize pruning's computational advantages, Anwar *et al*. [1] locate pruning candidates using particle filters in a structured way. Again, with each pruning candidate weighted separately, the across-layer relationship is largely ignored. Last but not least, particle filters are generally expensive considering the huge number of connections.

2.3. Alternatives to FC Layers as a Final Classifier

The FC layers, is basically a expensive final classifier (similar to logistic regression with a computationally intensive pre-transformation process) on top of the extracted CNN features. As such, this leads to the possibility that by replacing this layer with a different classifier, a reduction in computational complexity becomes possible. Many machine learning methods, including SVM have met with some success for classification tasks, including gender recognition [27]. An advantage of SVM over logistic regression is that different kernels enable SVM to deal with linearly inseparable data. As a result, a wide variety of methods have combined neural networks and SVMs [43, 38]. However, the reasoning behind the success of a particular combination is not usually provided. In [38], Sharif *et al*. combined CNN features and SVM for multiple visual recognition tasks and obtained state-of-the-art results. By replacing the softmax layer with a linear SVM and minimizing a margin-based loss, Tang [43] showed a small but consistent improvement on a variety of deep learning benchmark datasets such as MNIST and CIFAR-10. Another alternative is Bayesian classifier, which has a nice probabilistic interpretation similar to logistic regression but does not necessarily model the log-odds with a linear function. Due to its probabilistic nature, it may be optimal for challenging datasets with much noise and uncertainty. As demonstrated in [44], the Bayesian classifier can outperform SVM in gender recognition when there are a wide variety of occlusions and view changes present in the images.

3. Gender Classification Using a Deep but Light-weight Network

3.1. Network Structure

In this paper, our convolutional neural network is based on the very deep VGG-16 architecture [39] and is pre-trained using the ImageNet data in a similar way to [35]. The VGG-16 architecture is used as an example of a very deep network partly because its descendant, VGG-Face net [29], is experimentally testified to successfully learn discriminative facial features for face recognition. Simonyan and Zisserman [39] proposed the VGG-16 architecture with the claim that significant improvements can be gained by increasing the network depth while decreasing the filter sizes. In the VGG-16 model, 3x3 conv filters with stride 1 and pad 1, together with 2x2 max-pooling filters with stride 2 are employed. In order to simulate a larger filter size of 5*5 or 7*7, two or three conv layers are stacked together (before each pooling layer). This way the depth is increased without adding too many parameters. In our work, we fully train the network in the traditional manner before removing the FC layers, reducing the CNN feature dimensions, and plugging in alternative classifiers on top.

3.2. Dimension Reduction in the Last Conv Layer

Layer Conv5_3 is the last convolutional layer of the VGG net, which has 512 neurons. We define the maximum activation value of a neuron as its firing score. Then for each image a 512-D firing vector can be obtained in the last conv layer, which is called a firing instance or observation. By stacking all these observations extracted from a set of images, the firing data matrix X for that set is obtained. In our experiments, X is normalized as a pre-processing step. The benefits of abandoning less useful dimensions in X are twofold: 1) it compresses the data and thus has a potential for network pruning. 2) it can make the pattern hidden in the high dimensional data easier to find, which simplifies classification and possibly boost accuracy.

Principal Component Analysis (PCA) has been widely used for general dimensionality reduction based on total data variance. However, for supervised learning, compression should be based on the ability to discriminate between the (two) classes. Thus, PCA is no longer optimal because, without considering the labels, it may preserve unwanted variances while giving up discriminative information in low-variance dimensions. Inspired by Fisher's Linear Discriminant Analysis [8] and its applications on face images [4, 18, 3], we adopt the intra-class correlation (ICC) to better measure the information utility for gender recognition:

$$ICC = \frac{s^2(b)}{s^2(b) + s^2(w)} \tag{1}$$

where $s^2(w)$ is the variance within each gender, $s^2(b)$ is the variance between the two genders, and the sum of the two is the overall variance across all samples from both genders. When reducing dimension, we are trying to maximize ICC, which has an equal effect of maximizing the ratio of between-gender variance to within-gender variance. The direct multivariate generalization of it is:

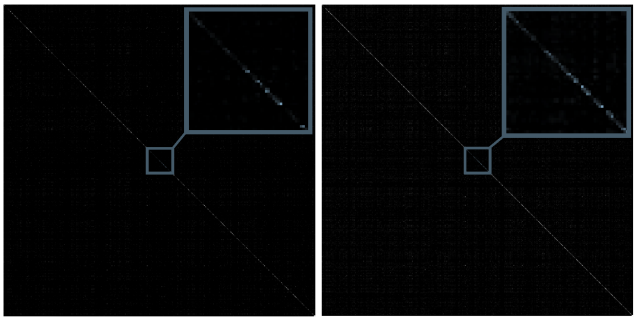
$$W_{opt} = \arg\max_{W} \frac{\mid W^T S_b W \mid}{\mid W^T S_w W \mid} \tag{2}$$

where

$$S_w = \sum_{i=0:1} \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T \tag{3}$$

$$S_b = \sum_{i=0:1} N_i (\mu_i - \mu)(\mu_i - \mu)^T \tag{4}$$

and W is the orthogonal transformation matrix projecting the data X from its original space to a new space with the columns in W being the new space's coordinate axes. x_k is a firing instance of the last conv layer, μ is the mean firing vector, and i indicates the gender (0 for female, 1 for male). Through analyzing S_w for both the LFW dataset and the CelebFaces Attributes Dataset (CelebA) [23] in our experiments, we find S_w tends to be a diagonal matrix (most large values are along the diagonal and most values off the diagonal have a zero or near zero value), which is to say, the firing of different neurons in the last conv layer is highly uncorrelated given the gender. Figure 1 shows the two S_w matrices of the training sets of LFW and CelebA. These results are intuitive given the fact that higher layers capture various high-level abstractions of the data (we have also examined other conv layers, the trend is that from bottom to top the neuron activations become progressively more decorrelated). Figure 2 shows some example Conv5_3 neuron patterns in our network trained on the CelebA dataset. Each pattern is synthesized via a regularized optimization algorithm [48] and can be interpreted as the pattern the corresponding neuron fires most on in the input image. Since the columns in W are the (generalized) eigenvectors of S_w (and S_b), W columns are the standard basis vectors and the elements on the diagonal of S_w (and S_b) are corresponding (generalized) eigenvalues. To maximize the ICC we simply need to select the neuron dimensions of low within-gender variance and high between-gender variance. For instance, although both the goatee neuron and the glasses neuron in Figure 2 have high variances (PCA prefers), the goatee dimension has a higher chance to be selected by LDA due to its higher ICC. This corresponds to intuition as most females do not have goatee while many males do. The direct abandonment of certain Conv5_3 neurons greatly facilitates the pruning at all other layers.



(a) S_w Matrix of LFW (b) S_w Matrix of CelebA

Figure 1: S_w matrices of (a) LFW and (b) CelebA. The one-pixel wide diagonals in both 512*512 matrices are best viewed when zoomed in (as demonstrated in blue squares).

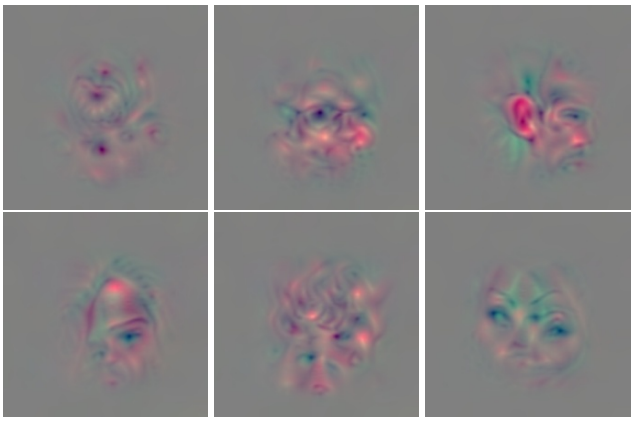


Figure 2: Sample Conv5_3 Neurons (trained On CelebA). From top left to bottom right, those neurons fire for goatee, glasses, ear, hairline, curly hair, and noses respectively.

3.3. Pruning of the Deep Network

Last conv layer dimensionality reduction along neuron directions makes pruning on the neuron (filter) level possible. With the removal of a filter, the dependencies of this filter on others in previous layers are also eliminated. When all the dependencies on a filter from higher layers are removed, this filter can be discarded. Take Figure 3 for example. The remaining filter outputs in a layer are colored in cyan. Corresponding useful depths of a next layer filter are colored in green (e.g. each useful $C3$ filter is represented by the small green block in column $C2$). The remaining cyan filter outputs/filters (overlapped with the green useful depths of a next layer filter) depend only on those cyan filter outputs/filters in the previous layer. Non-colored filter parts and filter outputs (filters) are thus discarded. When 106 $C2$ filters (each visualized by the small block in column $C1$) are thrown away, not only the $C2$ convolution computations with $C1$ output data are reduced by 106/128, but also $C3$ filters' depth is reduced by the same ratio (as shown in green in Column $C2$). The same applies when other layer filters are discarded. In total, 151,938 conv layer parameters are pruned away. In our work, the dependency of a filter on others in previous layers is calculated using deconv [50, 49], a technique mapping an activation through lower layers all the way to the pixel level. We choose de-

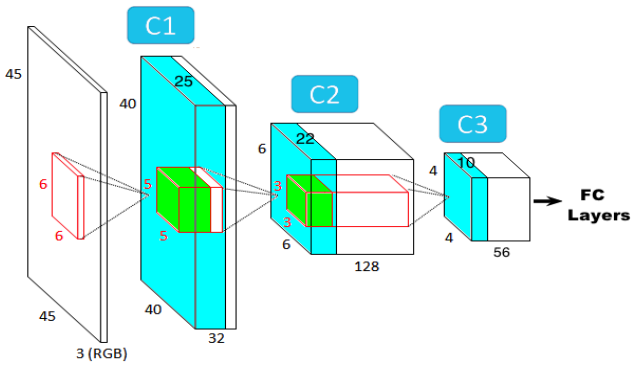


Figure 3: Demonstration of pruning on filter level (cyan indicates remaining data, green represents the surviving part of a remaining next layer filter).

conv over backprop for the reason that we only care about the maximum activation value of each neuron in the last conv layer. Also, deconv is more robust to noise and vanishing gradients. It is also worth noting that unlike traditional approaches, the dependency here is data-driven and pooled over training samples. Its improvement over weight-based pruning is due to the fact that neural networks are non-convex and trained weights are not guaranteed to be globally optimal. Therefore, a large weight does not always indicate high importance. Large weights connections that have never been activated on a task specific dataset are of little use for that task. This is especially true when the network is pre-trained for a different task and we do not have enough data when fine-tuning. When pruning, the neurons with a deconv dependency smaller than a threshold is deleted. In our experiments, such a threshold is not difficult to set. When the threshold is smaller than a certain value t_0 (e.g. when about 75% conv parameters are pruned away in the four Conv5_3 neurons case on LFW), an accuracy plateau is reached, beyond which point the accuracy does not change too much with the decrease of the threshold. t_0 is then selected as the threshold. This guarantees no accuracy loss during the pruning process. That said, if further pruning is required, the threshold on (the highest) deconv values can be increased at the risk of sacrificing accuracy. To recover high accuracy, retraining is needed after pruning. Otherwise, the accuracy could be greatly sacrificed. To leverage the previously learned network structure (co-adapted structures and primitive features in the first few layers), the pruned networks are retrained starting from the surviving parameters without re-initializing.

3.4. Alternative Classifiers on Top of CNN Features

As alternatives to the expensive FC layers, SVM (with linear and RBF kernels) and Bayesian quadratic discriminant analysis are explored in our experiments based on the reduced CNN features. SVM is a deterministic, discriminative classifier, which tries to fit a hyperplane between two classes with as wide a margin as possible. It focuses on samples near the margins but does not assign attention to others. The main advantage of SVM lies in its various kernels, which, when selected properly, empower SVM to perform well even for linearly inseparable tasks. On the other hand, the Bayesian classifier is a probabilistic, generative approach. Instead of just giving a binary choice, the Bayesian classifier is able to generate a probability distribution over all (not necessarily two) classes. In cases where many sources of noise and uncertainty exist and no separating hyperplane can be easily found, the Bayesian classifier may be a better choice than SVM. That said, non-naive Bayesian quadratic discriminant analysis is vulnerable to the curse of dimensionality.

4. Experiments and Results

Our programs are implemented with the help of the Caffe toolbox on an Nvidia Tesla K40 GPU and a quad-core Intel i7 CPU. We modified the Caffe source code by adding modules such as filter pruning and deconv dependency calculation. Using gender recognition as a particular example, we tested our nets on two popular face datasets.

4.1. Databases

In this paper, the annotated versions of two large-scale datasets, namely LFWA+ [23] and CelebA [23], are used.

The LFWA+ dataset is a richly labeled version of the popular Labeled Faces in the Wild (LFW) database [17], originally designed for face verification tasks. The dataset has 40 facial attributes labels (including gender) for each image and covers a large range of pose and background clutter variations. Some images even have multiple faces.

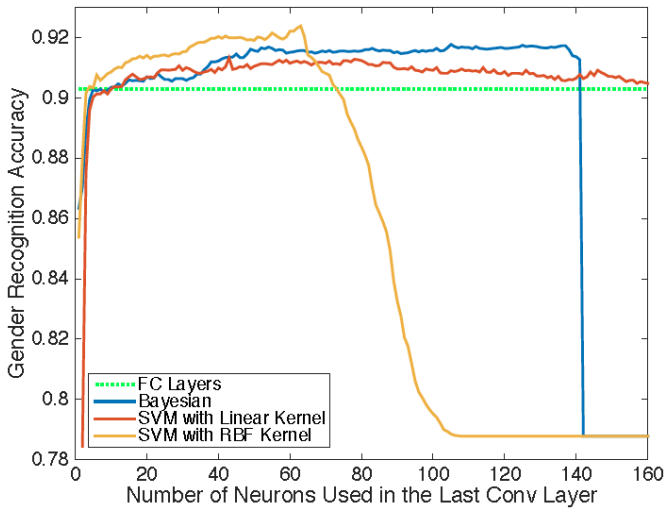
Another dataset used in our experiments is the CelebFaces Attributes Dataset (CelebA) [23], which is a large-scale dataset with 202,599 images of 10,177 identities, containing the same 40 facial attributes labels as in LFWA+. Despite its relatively large size, most of its images are portrait photos against simple backgrounds taken by professional photographers and are thus less challenging. For both databases, the train/test splits suggested in [23] are adopted. All the images are pre-resized to a dimension of 224*224.

4.2. Recognition Accuracy

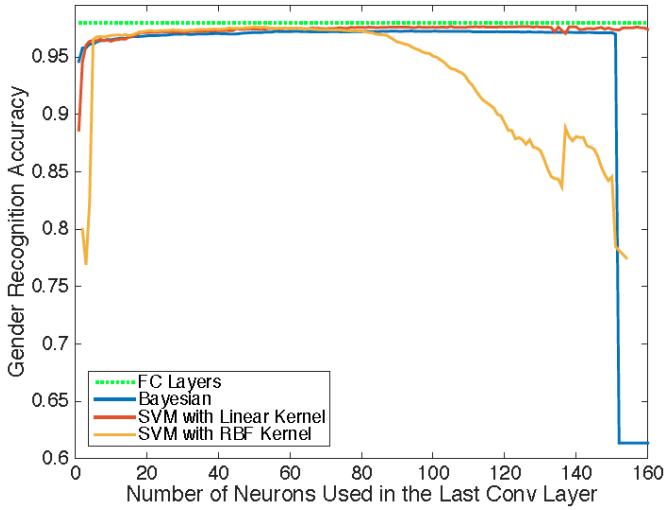
In this section, we demonstrate and discuss the recognition results of the Bayesian classifier and SVMs (Table 1) as well as their changes with the number of neurons preserved in the last conv layer on the LFW and CelebA datasets (Figure 4a and 4b). For comparison, the results of the original deep net are also included in the first row of Table 1 and as a green dashed line in Figure 4a and 4b. According to Table 1, both the Bayesian classifier and the SVMs can achieve higher or comparable (high) accuracies than the original net on the two datasets using only a small subset of

Method	LFW	CelebA
Original Net with FC	90.3% (512)	98.0% (512)
LDA-CNN+Bayesian	91.8% (105)	97.3% (94)
LDA-CNN+SVML	91.3% (43)	97.7% (105)
LDA-CNN+SVMR	92.4% (63)	97.5% (52)

Table 1: Highest recognition accuracy comparison of different approaches. SVML and SVMR represent SVM with linear and RBF kernel respectively. The accuracies reported here are the highest when a certain number (specified in the parentheses) of neurons are utilized in the last conv layer.



(a) Accuracy Comparison using Different Classifiers on LFW



(b) Accuracy Comparison using Different Classifiers on CelebA

Figure 4: Accuracy Comparison of Different Classifiers Based on Pruned CNN features

last conv layer neurons. Particularly, the Bayesian classifier and the SVM with RBF kernel (RBF-SVM) outperformed the original net by a margin of about 2% on LFW. Figure 4a and 4b show that the combination of only the 4 most discriminative Conv5_3 neurons (out of 512) can lead to high accuracies that are comparable to the full FC CNN. With more than four neurons, accuracies only improve slightly (<3%) with occasional decreases. This is consistent with our hypothesis that fine-tuned deep nets possibly have many less useful and redundant structures, which may sometimes hurt accuracy. On CelebA, the original FC CNN has comparable (slightly higher, within 1%) accuracies over the other LDA-CNN based classifications. In this case, most images are not as challenging as those in LFW, thus linear or generalized linear models (e.g. logistical regression based) are effectively able to separate the two classes. This can be seen as the linear SVM performs better than both RBF-SVM and the Bayesian classifier. Noticeably, the Bayesian classifier achieves a higher accuracy than linear SVM on the challenging LFW dataset where there is more uncertainty and noise. Also, on both datasets, the Bayesian classifier beats both SVMs when there are fewer than 3 neurons and has a more stable performance since it captures more information than just the margins. However, without the naive independence assumption of each dimension, the Bayesian classifier degrades drastically around 150 neurons due to the curse of dimensionality. The degradation kicks in suddenly as the space volume increases exponentially with dimensionality. Even one extra neuron dimension (e.g. from 150 to 151) can multiply the (already large) space volume. Since only a small number of neurons are needed, the Bayesian classifier is still a good choice, especially when memory resources are constrained. Although RBF-SVM performs well and attains the highest accuracy on LFW, it is slow and memory intensive to train on large datasets such as CelebA. In addition, compared to the Bayesian classifier, there are more parameters to set. Instead of choosing every parameter via cross validation, in our experiments, three sets of parameters are randomly selected for RBF-SVM and the accuracies reported here are comprised on their average output. Also, in both Figure 4a and 4b, the accuracy of the RBF-SVM kernel first increases and begins to decrease suddenly due to overfitting. This occurs a little later on CelebA than on LFW because of the former's larger size.

Compared to Bayesian and RBF-SVM, linear SVM performed similarly to the original FC classifier. This is intuitive in that FC layers, including softmax, is basically a logistic regression classifier with a transformed input and a linear SVM can be derived from logistic regression. Nevertheless, as will be shown in Subsection 4.4, the LDA-CNN-SVM structure is much more efficient than the original net.

4.3. Accuracy Change vs. Parameter Pruning Rate

In this subsection, we analyze the relationship of pruning rate and accuracy change in more details and compare

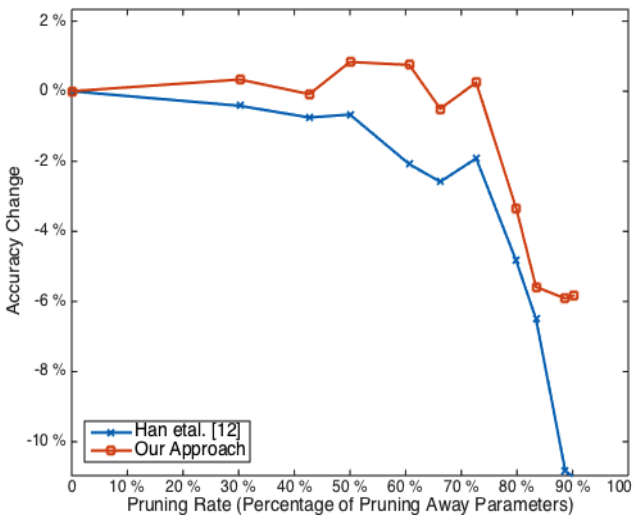


Figure 5: Accuracy change vs. conv layers pruning rate. Only 4 Conv5_3 neurons are used (when pruning rate$\neq$ 0).

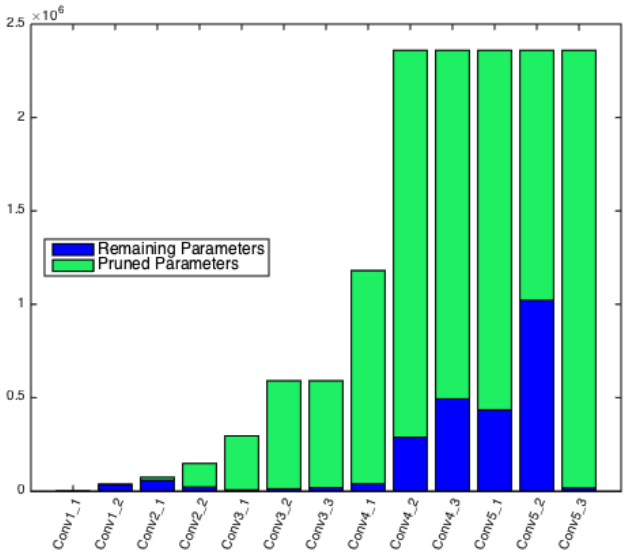


Figure 6: Demonstration of layerwise structure complexity reduction by keeping the 4 discriminative Conv5_3 neurons.

our results with a state-of-the-art pruning approach [12]. As shown in previous subsection, by preserving only four neurons in the last conv layer, we are able to achieve an accuracy comparable to the original DNN on the LFWA dataset. We take this case as an example. Figure 5 demonstrates its accuracy change/pruning rate relationship (by varying the dependency threshold). It is worth noting that unlike [12], we only retrain both pruned networks once. Additionally, for fair comparison, we also use (pruned) FC layers to classify our Fisher LDA reduced features. Since our goal is to prune CNN features for use with alternative lightweight classifiers, we keep the pruning rate of FC layers the same for both approaches and the pruning percentage reported is of only the conv layers. It is worth mentioning that all common hyper-parameters are set the same for both approaches and little tweaking of parameters is involved. That said, in order to escape local optima, smaller batch sizes are sometimes needed. As can be seen from Figure 5, our approach has higher accuracies across different pruning rates than [12]. At some points, accuracy can even improve slightly ($<$0.8%) when pruning due to the redundant and less useful structures in the hidden layers. Also, in our case, about only a quarter of all the conv layer weights are needed to maintain a comparable discriminating power. However, around 80%, both approaches suffer greatly from pruning. When the pruning rate reaches 84%, the accuracy of [12] keeps decreasing drastically while ours begins to be stable again. Our approach's better performance mainly stems from the awareness of each neuron's contribution to the final discriminating power when pruning the net. In other words, our approach's dependency is across all layers. On the other hand, the dependency in [12] is of length one. It may prune away small weights that contribute to more informative neurons in the last conv layer because the effect of small weights in a particular layer are possible to be accumulated over the layers or be enlarged by large weights in the following layers. Pruning away small weights in a certain layer is actually cutting off whole connections from the raw pixel level to the final classification stage. Even if weight magnitude is a good pruning measure, the importance of a whole bottom-to-top connection should not be measured by a length one weight. In next subsection, we will provide a computational complexity analysis in terms of both space and time.

4.4. Complexity Analysis

To gain more insight into our pruning method, Figure 6 and Table 2 offer a detailed layerwise space and time complexity analysis. According to Figure 6, most parameters in the middle conv layers (Conv2_2 to Conv4_1) do not help with our task. Compared to later layers, the first three layers have relatively low reduction rates. This is easy to understand given the observation that earlier layers contain more generic features such as edge and color blob detectors that could be useful to all classes. In addition, our approach's high pruning rate can directly contribute to lower memory requirements because unlike [12], it enables us to discard (rather than disregard) filter weights. In [12], masks are needed in the retraining stage to freeze zero weights. As a result, besides large overhead costs of extra masks, the number of convolutional operations does not actually change. That said, if masking is also applied in testing (at the cost of even more space), time would be saved since many multiplication operations are replaced by a simpler mask checking. Complexity can be further reduced if we replace the (pruned yet still large) FC layers with our lightweight al-

Method \ Layer	Conv1_1	Conv1_2	Conv2_1	Conv2_2	Conv3_1	Conv3_2	Conv3_3	Conv4_1
Original CNN+FC Layers	70.96	405.39	183.60	362.15	171.64	341.23	341.33	166.94
LDA-CNN+Bayesian/SVM	18.02	98.27	39.68	31.96	3.59	6.43	9.92	3.79
Speedup Ratio	3.93	4.13	4.63	11.33	47.83	53.06	34.41	44.08

Method \ Layer	Conv4_2	Conv4_3	Conv5_1	Conv5_2	Conv5_3	FC Layers			**Total**
						BC	SVML	SVMR	
Original CNN+FC Layers	333.75	333.98	85.69	85.70	85.63	283.20			**3306.50**
LDA-CNN+Bayesian/SVM	18.11	28.07	6.79	11.92	0.84	0.04	0.01	0.05	**286.86**
Speedup Ratio	18.43	11.90	12.63	7.19	101.68	7E3	3E4	6E3	**11.53**

Table 2: Per image recognition time comparison of different approaches in all layers (in milliseconds). BC is short for the Bayesian classifier, SVML and SVMR stand for SVM with linear and RBF kernel respectively. FC layers here refer to all the layers after Conv5_3. The tests are run on the CPU.

ternatives. Since our alternative classifiers are based only on the highest activation, they are more robust to noise (no performance degradation is incurred even when the FP16 precision is used). Compared to the original deep net model of 531 MB, our pruned model is very light and takes up only 7 MB (with no accuracy loss). For the Bayesian classifier, the storage overhead can be ignored when only four neurons are used (even when all neurons are utilized in Conv5_3, the extra space required is just about 2 MB). For SVMs, the extra storage needed depends on the number of trained support vectors. In the LFW and four neurons case, it is only about 30KB for both SVMs. Given the fact that most of today's latest cellphone models have only 1 or 2 GB RAM, the low storage requirements of our pruned nets are of great importance if we want to go from off-chip to on-chip.

Table 2 shows the recognition speed comparison between the original net and our pruned model. The original net is trained in the GPU mode using Caffe while tested with the CPU mode on. To avoid as much Caffe overhead as possible, we implement features extraction using survived filters ourselves utilizing all the four cores. According to the table, our LDA-Pruned model is faster at all conv layers than the original net. Besides the last conv layer, the middle layers with high structure reduction rates also enjoy a large speedup. Nonetheless, the relation is nonlinear owing to the different dimensions of each layer's input data. In total, an 11-fold speedup is achieved by using our pruned model. It is worth noting that both the SVMs and the Bayesian classifier (based on the reduced CNN features) are significantly faster than the original FC layers in classification. The Bayesian classifier's speed is somewhere between the two SVMs.

5. Discussion and Future Directions

While many big datasets are the property of large corporations (e.g. DeepFaces [42]), academic datasets are relatively small. Compact pruned nets like ours are easier to train and retrain, thus alleviating the data constraint to some extent and simultaneously improving on the generalization ability [21]. Furthermore, due to the low space and time complexity, pruned nets can possibly be embedded on the chip SRAM to deal with real-time video streams.

Although this paper targets gender classification, it is likely that the high decorrelation found in the last conv layer of our networks is common to other tasks and CNNs as well. However, more tests are needed for this to be seen. In the literature, classifications of different facial attributes are usually treated separately, despite the fact that they are related (males and females age, smile, and are perceived as attractive in different ways). As shown in Figure 2, when we train a deep net for gender classification, many local non identity related attributes such as nose, chin, and glasses are obtained in the last conv layer. One possible future research direction is then to tackle different facial traits in a more unified manner (possibly using only one DNN).

6. Conclusion

In this paper, we develop a deep but pruned CNN that can boost efficiency while maintaining accuracy in a facial trait classification task. In the gender classification example, a great many neuronal activations in the last conv layer were found to be highly uncorrelated within each gender. Through Fisher LDA, these neurons in dimensions that have low ICC were discarded, thereby greatly pruning the network and significantly increasing efficiency. As the result, the approach can be useful in contexts where fast and accurate performance is desirable but where expensive GPUs are not available (e.g. embedded systems). Our LDA based pruning is better than weight value based approaches because filter weights can be large but unimportant for the specific limited task when the pre-training is done on a large dataset of general recognition purposes (e.g. ImageNet). By combining with alternative classifiers, the approach is shown to achieve higher or comparable accuracies for gender recognition to the original deep net on the LFW and CelebA datasets, but with a reduction of model size by 70X, and with a subsequent 11-fold speedup.

References

[1] S. Anwar, K. Hwang, and W. Sung. Structured pruning of deep convolutional neural networks. *arXiv preprint arXiv:1512.08571*, 2015. 3

[2] S. Baluja and H. A. Rowley. Boosting sex identification performance. *International Journal of computer vision*, 71(1):111–119, 2007. 2

[3] J. Bekios-Calfa, J. M. Buenaposada, and L. Baumela. Revisiting linear discriminant techniques in gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):858–864, 2011. 3

[4] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997. 3

[5] C. Ben Abdelkader and P. Griffin. A local region-based approach to gender classification from face images. computer vision and pattern recognition-workshops, 2005. cvpr workshops. In *IEEE Computer Society Conference on. IEEE*, 2005. 2

[6] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007. 1

[7] E. Eidinger, R. Enbar, and T. Hassner. Age and gender estimation of unfiltered faces. *IEEE Transactions on Information Forensics and Security*, 9(12):2170–2179, 2014. 2

[8] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936. 1, 3

[9] B. A. Golomb, D. T. Lawrence, and T. J. Sejnowski. Sexnet: A neural network identifies sex from human faces. In *NIPS*, volume 1, page 2, 1990. 2

[10] S. Gutta and H. Wechsler. Gender and ethnic classification of human faces using hybrid classifiers. In *Neural Networks, 1999. IJCNN'99. International Joint Conference on*, volume 6, pages 4084–4089. IEEE, 1999. 2

[11] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *CoRR, abs/1510.00149*, 2, 2015. 2

[12] S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, pages 1135–1143, 2015. 1, 2, 7

[13] B. Hassibi and D. G. Stork. *Second order derivatives for network pruning: Optimal brain surgeon*. Morgan Kaufmann, 1993. 2

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 1, 2

[15] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. 1

[16] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. 1

[17] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 2, 5

[18] A. Jain, J. Huang, and S. Fang. Gender identification using frontal facial images. In *2005 IEEE International Conference on Multimedia and Expo*, pages 4–pp. IEEE, 2005. 2, 3

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

[20] 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(4):541–551, 1989. 1

[21] Y. LeCun, J. S. Denker, S. A. Solla, R. E. Howard, and L. D. Jackel. Optimal brain damage. In *NIPs*, volume 2, pages 598–605, 1989. 2, 8

[22] G. Levi and T. Hassner. Age and gender classification using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 34–42, 2015. 2

[23] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 2, 4, 5

[24] E. Makinen and R. Raisamo. Evaluation of gender classification methods with automatically detected and aligned faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):541–547, 2008. 2

[25] J. Mansanet, A. Albiol, and R. Paredes. Local deep neural networks for gender recognition. *Pattern Recognition Letters*, 70:80–86, 2016. 2

[26] C. P. MarcAurelio Ranzato, S. Chopra, and Y. LeCun. Efficient learning of sparse representations with an energy-based model. In *Proceedings of NIPS*, 2007. 1

[27] B. Moghaddam and M.-H. Yang. Learning gender with support faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):707–711, 2002. 2, 3

[28] A. J. O'toole, T. Vetter, N. F. Troje, and H. H. Bülthoff. Sex classification is better with three-dimensional head structure than with image intensity information. *Perception*, 26(1):75–84, 1997. 2

[29] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015. 3

[30] C. Perez, J. Tapia, P. Estévez, and C. Held. Gender classification from face images using mutual information and feature fusion. *International Journal of Optomechatronics*, 6(1):92–119, 2012. 2

[31] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss. The feret database and evaluation procedure for face-recognition algorithms. *Image and vision computing*, 16(5):295–306, 1998. 2

[32] B. Poggio, R. Brunelli, and T. Poggio. Hyberbf networks for gender classification. 1992. 2

[33] L. Y. Pratt. *Comparing biases for minimal network construction with back-propagation*, volume 1. Morgan Kaufmann Pub, 1989. 2

[34] D. Reid, S. Samangooei, C. Chen, M. Nixon, and A. Ross. Soft biometrics for surveillance: an overview. *Machine learning: theory and applications. Elsevier*, pages 327–352, 2013. 2

[35] R. Rothe, R. Timofte, and L. V. Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision (IJCV)*, July 2016. 2, 3

[36] G. Shakhnarovich, P. A. Viola, and B. Moghaddam. A unified learning framework for real time face detection and classification. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 14–21. IEEE, 2002. 2

[37] C. Shan. Learning local binary patterns for gender classification on real-world face images. *Pattern Recognition Letters*, 33(4):431–437, 2012. 2

[38] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014. 3

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 3

[40] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems*, pages 1988–1996, 2014. 2

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 1

[42] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014. 8

[43] Y. Tang. Deep learning using linear support vector machines. *arXiv preprint arXiv:1306.0239*, 2013. 3

[44] M. Toews and T. Arbel. Detection, localization, and sex classification of faces from arbitrary viewpoints and under occlusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1567–1581, 2009. 2, 3

[45] I. Ullah, M. Hussain, G. Muhammad, H. Aboalsamh, G. Bebis, and A. M. Mirza. Gender recognition from face images with local wld descriptor. In *19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 417–420. IEEE, 2012. 2

[46] A. Verma and L. Vig. Using convolutional neural networks to discover cogntively validated features for gender classification. In *Soft Computing and Machine Intelligence (ISCMI), 2014 International Conference on*, pages 33–37. IEEE, 2014. 2

[47] Z. Yang, M. Li, and H. Ai. An experimental study on automatic face gender classification. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 1099–1102. IEEE, 2006. 2

[48] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015. 4

[49] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014. 4

[50] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *2011 International Conference on Computer Vision*, pages 2018–2025. IEEE, 2011. 4