# Beyond Simple Features: A Large-Scale Feature Search Approach to Unconstrained Face Recognition

David Cox
The Rowland Institute at Harvard
Harvard University
Cambridge, MA 02142, USA
cox@rowland.harvard.edu

Nicolas Pinto
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
pinto@mit.edu

***Abstract*— Many modern computer vision algorithms are built atop of a set of low-level feature operators (such as SIFT [1], [2]; HOG [3], [4]; or LBP [5], [6]) that transform raw pixel values into a representation better suited to subsequent processing and classification. While the choice of feature representation is often not central to the logic of a given algorithm, the quality of the feature representation can have critically important implications for performance. Here, we demonstrate a large-scale feature search approach to generating new, more powerful feature representations in which a multitude of complex, nonlinear, multilayer neuromorphic feature representations are randomly generated and screened to find those best suited for the task at hand. In particular, we show that a brute-force search can generate representations that, in combination with standard machine learning blending techniques, achieve state-of-the-art performance on the *Labeled Faces in the Wild (LFW)* [7] unconstrained face recognition challenge set. These representations outperform previous state-of-the-art approaches, in spite of requiring less training data and using a conceptually simpler machine learning backend. We argue that such large-scale-search-derived feature sets can play a synergistic role with other computer vision approaches by providing a richer base of features with which to work.**

## I. Introduction

Face recognition has long been, and continues to be, a highly active area of research [8], [9], [10], [11], [12], [13], [14], [15], [14], [16], [17]. In recent years, interest in the problem of *unconstrained* face recognition has grown in the community, driven in large part by the creation of the *Labeled Faces in the Wild (LFW)* [7] test set, which has provided a standardized benchmark against which to measure progress. While face recognition research *per se* has a long and rich history, much work prior to the last decade was focused on face recognition in relatively constrained environments (e.g. posed photographs, under controlled lighting conditions [18], [19], [20], [21], [22], [23]). More recently, thanks in large part to the rise of the internet, it has become possible to assemble large collections of face images "in the wild" in the sense that they come from a wide variety of sources and were not posed for the purpose of research. While this set has proven to be quite challenging, large strides have been in made in recent years towards higher performance [24], [25], [26], [27], [28], [29].

While a variety of different approaches to the *LFW* set have been taken, a common feature of most approaches is the use of some low-level visual feature set, such as SIFT [1], [2]; HOG [3], [4]; or LBP [5], [6] that transforms raw pixels values into a better form for subsequent processing. While individual algorithms often do not depend critically on the choice of a particular feature representation used, the choice of features used does frequently play a key role in determining performance. Meanwhile, there are only a handful of visual feature representations in common use, and arguably less attention has been paid to developing new or better features.

One potentially promising source for new, more complex visual feature representations is the class of "biologically-inspired" representations. Biologically-inspired approaches seek to build artificial visual systems that capture aspects of the computational architecture of the brain, in the hope of eventually mimicking its computational abilities. Such efforts to model visual computations done by the brain have a long history, at least dating back to Fukushima's Neocognitron (1980; [30]). More recent experiments with biologically-inspired models have shown them to be highly competitive in a variety of different face and object recognition contexts [31], [32], [33], [24], [34].

However, the range of possible feature representations that would count as "biologically-inspired" is broad, and it is not clear which particular instantiations of biologically-inspired ideas are best for a given task. Pinto et al. [35] previously demonstrated a high-throughput screening approach for biologically-inspired algorithms, wherein a large number of possible candidate models from an inclusive model family are considered, and the best performing models are "skimmed off the top" and evaluated further. However, while that work showed success with synthetic test images, it has not been known to date whether models from this class are competitive with current state-of-the-art approaches on standard face and object recognition test sets.

Here we present a modified large-scale feature search procedure that simplifies and accelerates the search procedure described in [35], with the goal of generating feature representations tailored for unconstrained face recognition, as embodied by the *LFW* test set. Multiple complimentary representations are further derived through training set augmentation, alternative face comparison functions, and feature set searches with a varying number of model layers. These individual feature representations are then combined using kernel techniques to achieve even better performance. We show that our approach yields multiple feature sets that

outperform previous state-of-the-art approaches on the *LFW* set, even while requiring less training data and using simpler machine learning backends. In addition to providing evidence for the utility of large-scale feature search for standard "real world" test sets, these results emphasize the value of good underlying representations and point a path forward in the generation of new, more powerful visual features.

## II. Methods

### A. Large-scale feature search framework

The large-scale feature search approach used here consists of four basic components: (1) a parametric family of feature representation, wherein key aspects of the behavior of the features are controlled by a fixed set of parameters, (2) a generation procedure for choosing models from the larger family to evaluate, (3) a screening procedure, run on each candidate feature representation, to determine which models to evaluate further and (4) a validation procedure, using independent data, to evaluate the utility of representations found during the screening procedure.

The approach we follow here is similar to that described in [35], with two important differences, which we describe briefly here, and detail in depth below. First, Pinto et al. [35] used an unsupervised learning procedure in order to learn certain model parameters from a pre-training video set. Here, we dispense with this unsupervised learning procedure, instead opting for greatly speeded model generation, allowing more model architectures to be evaluated per unit time. Second, we used the *LFW View 1* subset as a screening set. Details of the model family considered, and generation, screening and validation procedures used are described below.

### B. Biologically-inspired visual representations

In our experiments, we used two basic classes of biologically-inspired visual representations, shown in Fig. 1.

First, as a control, we used *V1-like*, a one-layer model characterized by a cascade of linear and nonlinear processing steps and designed to encapsulate some of the known properties of the first cortical processing stage in the primate brain. Our *V1-like* implementation was taken without modification from [33], [24].

Second, we used two and three layer models following the basic multi-layer model scheme described in [35]. Briefly, these models consist of multiple stacked layers of linear-nonlinear processing stages, similar to those in the *V1-like* model. Importantly, in order to speed the processing of these models, we disabled the unsupervised learning mechanisms described in [35] and instead used *random* filter kernels drawn from a uniform distribution. Prior experience of our group and others [34] has suggested that random filters can in many cases function surprisingly well for models belonging to this general class. Details of each model class follow.

### C. "V1-like" visual representation

In the *V1-like* representation, features were taken without additional optimization from Pinto et al.'s *V1S+* [33]. This visual representation is based on a first-order description of primary visual cortex V1 and consists of a collection of locally-normalized, thresholded Gabor wavelet functions spanning a range of orientations and spatial frequencies.

In spite of their simplicity, these features have been shown to be among the best-performing non-blended features set on standard natural face and object recognition benchmarks [33], [24], [25] (i.e. *Caltech-101*[36], *Caltech-256*[37], *ORL*[18], *Yale*[19], *CVL*[20], *AR*[21], *LFW*[7]) and are a key component of the best blended solutions for some of these same benchmarks [38]. We used the authors' publicly available source code to generate these features and followed the same basic read-out/classification procedure as detailed in [33], with two minor modifications. Specifically, no PCA dimensionality reduction was performed prior to classification (the full vector was used), and a different SVM regularization parameter was used ($C = 10^5$ instead of $C = 10$, see below).

For a detailed description of the *V1-like* visual representation, we refer the interested reader to the methods of the original publication [33] and its source code.

### D. High-throughput-derived multilayer visual representations: HT-L2 *and* HT-L3

*1) Model architecture: :* Candidate models were composed of a hierarchy of two (*HT-L2*) or three layers (*HT-L3*), with each layer including a cascade of linear and nonlinear operations that produce successively elaborated nonlinear feature-map representations of the original image. A diagram detailing the flow of operations is shown in Fig. 1, and, for the purposes of notation, the cascade of operations is represented as follows:

$$Layer^0 : \quad \mathbf{Input} \overset{\mathbf{Grayscale}}{\longrightarrow} \overset{\mathbf{Normalize}}{\longrightarrow} \mathbf{N^0}$$

and generally, for all $\ell \geq 1$:

$$Layer^\ell : \quad \mathbf{N}^{\ell-1} \overset{\mathbf{Filter}}{\longrightarrow} \mathbf{F}^\ell \overset{\mathbf{Activate}}{\longrightarrow} \mathbf{A}^\ell \overset{\mathbf{Pool}}{\longrightarrow} \mathbf{P}^\ell \overset{\mathbf{Normalize}}{\longrightarrow} \mathbf{N}^\ell$$

Details of these steps along with the range of parameter values included in the random search space are described next.

*2) Input and Pre-processing:* The input of the *HT-L2* and *HT-L3* models were 100x100 and 200x200 pixel images, respectively. In the pre-processing stage, referred to as $Layer^0$, this input was converted to grayscale and locally normalized:

$$\mathbf{N^0} = \mathbf{Normalize}(\mathbf{Grayscale}(\mathbf{Input})) \qquad (1)$$

where the **Normalize** operation is described in detail below. Because this normalization is the final operation of each layer, in the following sections, we refer to $N^{\ell-1}$ as the input of each $Layer^{\ell>0}$ and $N^\ell$ as the output.

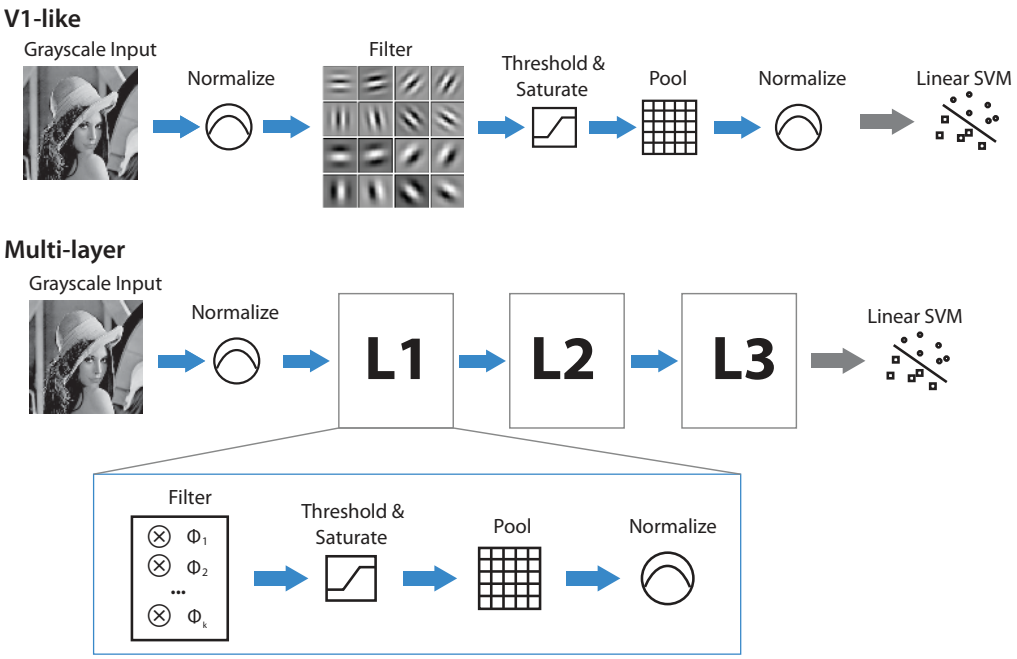



Fig. 1. **A schematic diagram of the system architecture of the family of models considered.** Each model consists of one to three feedforward filtering layers, with the filters in each layer being applied across the previous layer.

*3) Linear Filtering:* The input $N^{\ell-1}$ of each subsequent layer (i.e. $Layer^{\ell}, \ell \in \{1,2,3\}$) was first linearly filtered using a bank of $k^{\ell}$ filters to produce a stack of $k^{\ell}$ feature maps, denoted $F^{\ell}$. In a biologically-inspired context, this operation is analogous to the weighted integration of synaptic inputs, where each filter in the filterbank represents a different cell.

The filtering operation for $Layer^{\ell}$ is denoted:

$$\mathbf{F}^{\ell} = \mathbf{Filter}(\mathbf{N}^{\ell-1}, \mathbf{\Phi}^{\ell}) \tag{2}$$

and produces a stack, $F^{\ell}$, of $k^{\ell}$ feature maps, with each map, $F_i^{\ell}$, given by:

$$F_i^{\ell} = N^{\ell-1} \otimes \Phi_i^{\ell} \quad \forall i \in \{1,2,\ldots,k^{\ell}\} \tag{3}$$

where $\otimes$ denotes a correlation of the output of the previous layer, $N^{\ell-1}$ with the filter $\Phi_i^{\ell}$ (e.g. sliding along the first and second dimensions of $N^{\ell-1}$). Because each successive layer after $Layer^0$ is based on a stack of feature maps, $N^{\ell-1}$ is itself a stack of 2-dimensional feature maps. Thus, the filters contained within $\Phi^{\ell}$ are, in turn, 3-dimensional, with the their third dimension matching the number of filters (and therefore, the number of feature maps) from the previous layer (i.e. $k^{\ell-1}$).

*Parameters:*

- The filter shapes ${f_s}^{\ell} \times {f_s}^{\ell} \times {f_d}^{\ell}$ were chosen randomly with ${f_s}^{\ell} \in \{3,5,7,9\}$ and ${f_d}^{\ell} = k^{\ell-1}$.
- Depending on the layer $\ell$ considered, the number of filters $k^{\ell}$ was chosen randomly from the following sets:
  - In $Layer^1$, $k^1 \in \{16,32,64\}$
  - In $Layer^2$, $k^2 \in \{16,32,64,128\}$
  - In $Layer^3$, $k^3 \in \{16,32,64,128,256\}$

All filter kernels were fixed to random values drawn from a uniform distribution.

*4) Activation Function:* Filter outputs were subjected to threshold and saturation activation function, wherein output values were clipped to be within a parametrically defined range. This operation is analogous to the spontaneous activity thresholds and firing saturation levels observed in biological neurons.

We define the activation function:

$$\mathbf{A}^{\ell} = \mathbf{Activate}(\mathbf{F}^{\ell}) \tag{4}$$

that clips the outputs of the filtering step, such that:

$$\mathbf{Activate}(\mathbf{x}) = \begin{cases} {\gamma_{max}}^{\ell} & \text{if } x > {\gamma_{max}}^{\ell} \\ {\gamma_{min}}^{\ell} & \text{if } x < {\gamma_{min}}^{\ell} \\ x & \text{otherwise} \end{cases} \tag{5}$$

Where the two parameters ${\gamma_{min}}^{\ell}$ and ${\gamma_{max}}^{\ell}$ control the threshold and saturation, respectively. Note that if both minimum and maximum threshold values are $-\infty$ and $+\infty$, the activation is linear (no output is clipped).

*Parameters:*

- ${\gamma_{min}}^{\ell}$ was randomly chosen to be $-\infty$ or 0
- ${\gamma_{max}}^{\ell}$ was randomly chosen to be 1 or $+\infty$

*5) Pooling:* The activations of each filter within some neighboring region were then pooled together and the resulting outputs were spatially downsampled.

We define the pooling function:

$$\mathbf{P}^{\ell} = \mathbf{Pool}(\mathbf{A}^{\ell}) \tag{6}$$

such that:

$$\mathbf{P}_{\mathbf{i}}^{\ell} = \mathbf{Downsample}_{\alpha}(\sqrt[p^{\ell}]{(A_i^{\ell})^{p^{\ell}} \odot \mathbf{1}_{a^{\ell} \times a^{\ell}}}) \tag{7}$$

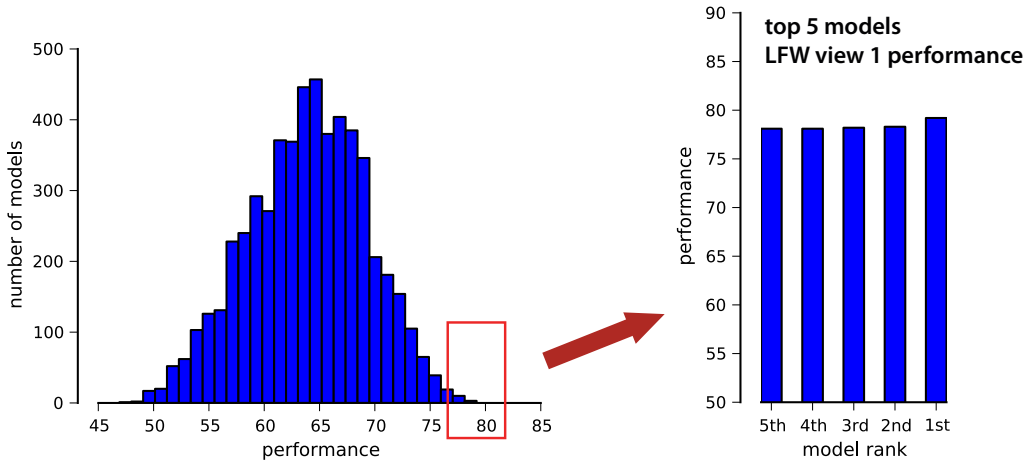



Fig. 2. **The high-throughput screening process used to find good representations.** Here, data is shown for the screening of *HT-L3* models. A distribution of the performance of 6,917 randomly generated models is shown on the left, with the top five high-performing models replotted on the right. Following screening, the models were evaluated exclusively with sets that do not overlap with the screening set.

, where $\odot$ is the 2-dimensional correlation function with $\mathbf{1}_{a^\ell \times a^\ell}$ being an $a^\ell \times a^\ell$ matrix of ones ($a^\ell$ can be seen as the size of the pooling "neighborhood"). The variable $p^\ell$ controls the exponents in the pooling function.

*Parameters:*

- The stride parameter $\alpha$ was fixed to 2, resulting in a downsampling factor of 4.
- The size of the neighborhood $a^\ell$ was randomly chosen from $\{3, 5, 7, 9\}$.
- The exponent $p^\ell$ was randomly chosen from $\{1, 2, 10\}$.

Note that for $p^\ell = 1$, this is equivalent to blurring with a $a^\ell \times a^\ell$ boxcar filter. When $p^\ell = 2$ or $p^\ell = 10$ the output is the $L^{p^\ell}$-norm [1].

*6) Normalization:* As a final stage of processing within each layer, the output of the Pooling step was normalized by the activity of their neighbors within some radius (across space and across feature maps). Specifically, each response was divided by the magnitude of the vector of neighboring values if above a given threshold. This operation draws biological inspiration from the competitive interactions observed in natural neuronal systems (e.g. contrast gain control mechanisms in cortical area V1, and elsewhere [39], [40])

We define the normalization function:

$$\mathbf{N}^\ell = \mathbf{Normalize}(\mathbf{P}^\ell) \tag{8}$$

such that:

$$N^\ell = \begin{cases} \rho^\ell \cdot C^\ell & \text{if } \rho^\ell \cdot \left\| C^\ell \otimes \mathbf{1}_{b^\ell \times b^\ell \times k^\ell} \right\|_2 < \tau^\ell \\ \dfrac{C^\ell}{\left\| C^\ell \otimes \mathbf{1}_{b^\ell \times b^\ell \times k^\ell} \right\|_2} & \text{otherwise} \end{cases} \tag{9}$$

with

$$C^\ell = P^\ell - \delta^\ell \cdot \frac{P^\ell \otimes \mathbf{1}_{b^\ell \times b^\ell \times k^\ell}}{b^\ell \cdot b^\ell \cdot k^\ell} \tag{10}$$

Where $\delta^\ell \in \{0, 1\}$, $\otimes$ is a 3-dimensional correlation over the "valid" domain (i.e. sliding over the first two dimensions only), and $\mathbf{1}_{b^\ell \times b^\ell \times k^\ell}$ is a $b^\ell \times b^\ell \times k^\ell$ array full of ones. $b^\ell$ can be seen as the normalization "neighborhood" and $\delta^\ell$ controls if this neighborhood is centered (i.e. subtracting the mean of the vector of neighboring values) before divisive normalization. $\rho^\ell$ is a "magnitude gain" parameter and $\tau^\ell$ is a threshold parameter below which no divisive normalization occurs.

*Parameters:*

- The size $b^\ell$ of the neighborhood region was randomly chosen from $\{3, 5, 7, 9\}$.
- The $\delta^\ell$ parameter was chosen from $\{0, 1\}$.
- The vector of neighboring values could also be stretched by gain values $\rho^\ell \in \{10^{-1}, 10^0, 10^1\}$. Note that when $\rho^\ell = 10^0 = 1$, no gain is applied.
- The threshold value $\tau^\ell$ was randomly chosen from $\{10^{-1}, 10^0, 10^1\}$.

### *E. Final model output dimensionality*

The output dimensionality of each candidate model was determined by the number of filters in the final layer, and the x-y "footprint" of the layer (which, in turn, depends on the subsampling at each previous layer). In the model space explored here, the possible output dimensionality ranged from 256 to 73,984.

### *F. Screening (model selection)*

A total of 5,915 *HT-L2* and 6,917 *HT-L3* models were screened on the *LFW* View 1 "aligned" set [26]. We selected the best five models from each "pool" for further analysis on the *LFW* View 2 set (Restricted Protocol). Note that *LFW* View 1 and View 2 do not contain the same individuals and are thus mutually exclusive sets. View 1 was designed as a model selection set while View 2 is used as an independent validation set for the purpose of comparing different methods.

Examples of the screening procedure for *HT-L2* and *HT-L3* models on the *LFW* View 1 task screening task are shown

[1]The $L^{10}$-norm produces outputs similar to a *max* operation (i.e. *softmax*).

in Fig. 2. Performance of randomly generated *HT-L3* models ranged from chance performance (50%) to better than 80% correct; the best five models were drawn from this set and are denoted *HT-L3-1st*, *HT-L3-2nd*, and so on. An analogous procedure was undertaken to generate five two-layer models, denoted *HT-L2-1st*, *HT-L2-2nd*, etc.

### G. Evaluation Protocol

To evaluate the performance of our biologically-inspired representations, we followed the standard *LFW* face verification "Restricted View 2" protocol. 6,000 different face image pairs (half "same", half "different") were drawn randomly from the sets and divided into 10-fold cross validation splits with 5,400 training and 600 testing examples each.

Because the biologically-inspired representations used here generate one feature vector per image, comparison functions were used to generate a new feature vector for each pair, and these "comparison" features were used to train binary ("same" / "different") hard-margin linear SVM classifiers. Following [25] we used the following element-wise comparison functions: $|F_1 - F_2|$, $\sqrt{|F_1 - F_2|}$, $(F_1 - F_2)^2$, where $F_1$ and $F_2$ are the feature vectors generated from the first and the second image of the pair, respectively. We additionally added the comparison function $(F_1 \cdot F_2)$, which was not used in [25], under the logic that it serves as a soft "AND"-like function (i.e. it primarily results in a large response for elements where both $F_1$ and $F_2$ are large). We hypothesized that such a function would be valuable since our representations are all quite sparse, and thus a coincidence of high feature values in common between the two test images is likely to provide meaningful evidence of similarity.

### H. Kernel combinations and data-set augmentation

While the high-throughput search techniques described above are capable of yielding relatively high-performing individual representations for *LFW* by themselves, effectively all of the top-performing face recognition systems on *LFW* employ some form of more advanced machine learning backend to enhance their performance [26], [27], [28], [29]. One common approach in this regard is to blend together a large number of weak learners to produce a blended classifier.

To explore what performance enhancement can be gained with modest amounts of blending on top of our feature representations, we pursued a progressive strategy of layering on additional kernels to produce successively larger and higher performing blends. Two basic strategies were used for generating new kernels: 1) feature augmentation, performing operations on the input image, such as cropping and rescaling to produce alternate kernels using the same representation, and 2) representation blending, that is, combining together kernels derived from multiple separate feature representations (e.g. blending over the five *HT-L2* top models, or combining the top five *HT-L2* and *HT-L3* models).

The progression of these additional elaborations is described below:

*1) Multiple rescaled crops:* Following [25], we augmented the dataset by computing features on three different centered crops of the image: 250x250 (original), 150x150 and 125x75. Each of these crops was resized to the standard input size of each representation, and SVMs were trained separately for each crop size. Blending of the resulting kernels was done by simple kernel addition, with each kernel being trace-normalized (by the training kernel trace) prior to summation. More sophisticated blending (e.g. IKL/MKL[41], LP-Boost[38]) were not used at this stage.

*2) Blending of the top 5 models within class:* While the top five models found by our high-throughput search all yield similar levels of performance, they achieve this performance with different parameter sets. Consequently, to the extent that the top five models represent a diversity of different ways to achieve good performance, we would expect that blending these models would yield further enhancement of performance. At this stage, we combined all of the Stage 1 kernels above (multiple rescaled crops) from each of the top five models within each model-class (e.g. *HT-L2* and *HT-L3*).

*3) Hierarchical blends across model class:* Finally, we also explored a more principled way to blend the representations from each model class. Following [42] we assigned exponentially larger weight to higher-level representation (*V1-like* < *HT-L2* < *HT-L3*) resulting in the following kernel:

$$K(\cdot,\cdot) = \sum_{\ell} (2^{\ell-1}) k_{\ell}(\cdot,\cdot) \tag{11}$$

where $\ell = 1$ for *V1-like* (one layer), $\ell = 2$ for the top five *HT-L2* (two layers) and $\ell = 3$ for the top five *HT-L3* (three layers).

We note that the choice of blending strategies to consider on the View 2 set was driven by performance on the View 1 set, thereby avoiding selection bias artifacts.

## III. Results

### A. High-throughput screening with LFW View 1

Fig. 2 shows the results of high-throughput screening to select model instantiations that are well-suited to the *LFW* verification task. For each model class, a multitude of models were randomly generated and evaluated on the *LFW* view 1 set, and the best five were selected for further analysis.

### B. Performance on LFW Restricted View 2

Performance of individual models and model blends are shown in Table I. Performance ranging from 77.1 % for the simplest *V1-like* model to 88.1% for the largest blend were observed. Taken together, these results show that state-of-the-art level performance is possible within the model family, and there exist multiple paths (e.g. based purely on *V1-like* models, and based on high-throughput, multi-layer models) to achieving high levels of performance. Fig. 3 shows receiver-operator characteristic (ROC) curves for each of these models.

Interestingly, the inclusion of a single additional comparison function to the *V1-like* model blend described in [25] brings an additional 3% performance, placing it close to

TABLE I

PERFORMANCE (*LFW* RESTRICTED VIEW 2) OF THE FAMILY OF BIOLOGICALLY-INSPIRED MODELS AND BLENDS THEREOF.

| | alone | +crops | within blend | V1+L2+L3 | V1+L2+L3$_{(\text{weighted})}$ |
|---|---|---|---|---|---|
| ***V1-like*** | 77.0 ± 0.5 | 82.4 ± 0.5 | | | |
| ***HT-L2*** | | | | | |
| 5th | 77.8 ± 0.4 | 82.8 ± 0.5 | | | |
| 4th | 81.3 ± 0.4 | 85.4 ± 0.6 | | | |
| 3rd | 81.5 ± 0.6 | 85.1 ± 0.5 | 87.5 ± 0.5 | | |
| 2nd | 80.8 ± 0.4 | 83.6 ± 0.5 | | | |
| 1st | 81.0 ± 0.3 | 83.3 ± 0.5 | | 87.6 ± 0.6 | 88.1 ± 0.6 |
| ***HT-L3*** | | | | | |
| 5th | 82.8 ± 0.6 | 84.5 ± 0.6 | | | |
| 4th | 82.3 ± 0.3 | 82.7 ± 0.5 | | | |
| 3rd | 83.3 ± 0.4 | 85.6 ± 0.6 | 87.8 ± 0.4 | | |
| 2nd | 83.9 ± 0.3 | 86.8 ± 0.4 | | | |
| 1st | 84.1 ± 0.3 | 86.8 ± 0.3 | | | |

the last reported best performance on this set, even without extensive blending. Furthermore, we see that individual *HT-L3* models also perform surprisingly well — coming to within a few percent correct of the previous state-of-the-art.

A major advantage of our high-throughput approach is that it produces not one, but a diversity of models, and this situation is ideally suited to kernel blending approaches. Once blending is added, especially when coupled with an intelligent algorithm for weighting blended kernels, several different blends achieved performance exceeding previously reported state-of-the-art values (see Figure 3(c)). ROC curves for various blend groupings are shown in Fig. 3.

### C. Analysis of Errors

To understand better where room for improvement lies, we examined the error trials (misses and false alarms) produced by each model for quantitative and qualitative trends. To determine whether different models were primarily making the same or different errors, we segregated the responses of the *V1-like* and *HT-L3* models (rescaled-crop augmented variants, see Methods) into four categories: hits, misses, false positives, and correct rejections. We then computed the fraction of errors that these two models held in common and found 84.3% of false positives were the same across the two models, and that 87.3% of misses were missed by both models. This high level of consistency between error cases across the two models led us to ask whether a subset of "hard" images within the larger *LFW* set could be driving errors and capping performance.

Fig. 4 shows examples of misses and false positives held in common for both models. While developing a quantitative framework within which to analyze these errors is beyond the scope of this paper, several patterns are evident, even upon casual inspection. First, misses are dominated by situations where the individual-to-be-matched is seen in non-frontal view in at least one of the images. Second, false positives appear to occur more often in cases where different individuals appear in a very similar view, or with a similar expression.

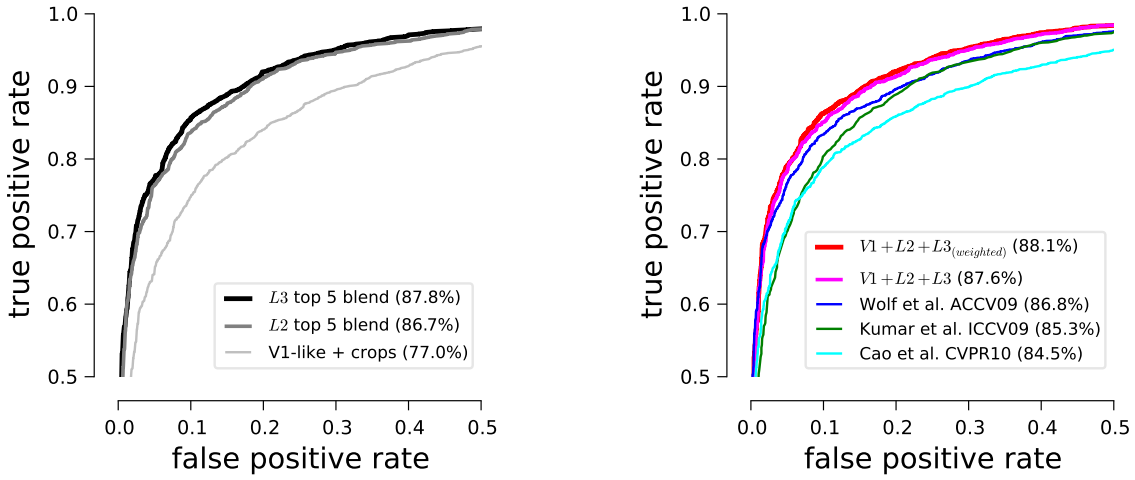
## IV. DISCUSSION

Our results provide more evidence that biologically-inspired models represent a promising and powerful direction in face recognition research. Individual models from this class are able to achieve good performance (e.g. around 77% for *V1-like* models, 84% for *HT-L3*), and blends of these models achieve more than 88% correct performance, beating previously reported state-of-the-art values.

Consistent with expectations, progressively more complex, multi-layer models are able to outperform the simpler *V1-like* model. Whether this higher performance is due to a greater ability to tolerate image variation (one of the original purposes for the construction of the *HT-L3* model class[35]) or some other factor remains to be seen. It should be noted that the *HT-L2* and *HT-L3* models used here were substantially simplified from those present in [35], in that they did not have structured filter kernels, nor were they subjected to any unsupervised learning. Whether adding these features back will result in higher levels of performance is an important future research question.

While there still remains substantial room for improvement, concerns that the *LFW* set does not necessarily accurately reflect the "full" problem of unconstrained face recognition remain [24], [25], [28]. *LFW* includes only a handful of examples per individual, and these photographs were often taken in the same setting and at the same event. Furthermore, Kumar et al. [28] showed that human observers were able to perform at greater-than-90% correct even when the faces themselves were masked out of the test images, indicating that the backgrounds in the *LFW* are more than sufficient for solving the task at a level higher than the current machine state of the art.

An analysis of the errors made by our models provides some clues about which parts of the *LFW* set are difficult and which ones are not. Our models failed on remarkably similar sets of face pairs, indicating that a common core of "hard" images may exist within the larger *LFW* set. A striking, albeit anecdotal, observation is that common error cases are dominated by misses when the same individual is shown in




(a) within-model-class blends (b) across-model-class blends

| Reference | Kumar et al. ICCV09 [28] | Wolf et al. ACCV09 [27] | Cao et al. CVPR10 [29] | This paper |
|---|---|---|---|---|
| Mean classification error | 14.7%±1.2 | 13.2%±0.3 | 15.5%±0.5 | **11.9%±0.6** |

(c) comparison with literature

Fig. 3. **ROC curves for various model sub-families on *LFW* Restricted View 2.** Curves for [27], [28] and [29] are plotted in 3(b) for reference. Plots are zoomed-in to facilitate comparison.

(a) Misses (b) False Positives

Fig. 4. **Examples of common errors across models.** Misses tend to be dominated by differences in view, while false positives frequently occur when different individuals share a common view or expression.

differing views and by false positives when two different individuals are compared while viewed from a similar angle (e.g. Fig. 4). An important feature of the *LFW* set is that faces must be detected by a Viola-Jones face detector in order to be included in the set, and this effectively restricts the range of face views that enter into the set (i.e. there is a bias towards frontal views). We hypothesize that those more off-axis views that do manage to pass the face detection filter will present a particularly difficult challenge for a system trained on the *LFW* set. The low-level (e.g. pixel-level) difference between two different views of the same individual can easily be larger than the low-level differences between two individuals

in a similar pose. A system that is not specially designed to tolerate this kind of variation will have a high false alarm rate on trials where two different individuals are seen in the same pose and a high miss rate where the same individual is compared across different poses. At the same time, if the *LFW* set contains a relatively small fraction of these off-axis faces, then a system trained exclusively on the *LFW* set will face difficulty learning to tolerate these cases, even if that system has the capability to learn such tolerance in principle.

As continued research manages to chip away at the remaining "performance gap" between human and machines on the *LFW* set, increased attention will need to be paid to whether *LFW* truly represents the problem of interest. On one hand, as long as some performance gap exists, the set is obviously valid at a basic level. However, the question remains whether a "fuller" formulation of the problem (i.e. more natural, less filtered) might lead to faster progress.

## V. Acknowledgments

The authors would like to thank Hanspeter Pfister, Wen-Mei Hwu, Volodymyr Kindratenko and Jeremy Enos for making additional GPU clusters available for this work. This work was funded by the Rowland Institute of Harvard, the NVIDIA Graduate Fellowship, and the National Science Foundation (IIS 0963668). Hardware support was generously provided by the NVIDIA Corporation.

## References

[1] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
[2] Luo, J., Ma, Y., Takikawa, E., Lao, S., Kawade, M., Lu, B.: Person-specific SIFT features for face recognition. ICASSP (2007)
[3] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. CVPR (2005)
[4] Albiol, A., Monzo, D., Martin, A., Sastre, J., Albiol, A.: Face recognition using HOG-EBGM. Pattern Recognition Letters (2008)
[5] ahonen, t., hadid, a., pietikainen, m.: face recognition with local binary patterns. eccv (2004)
[6] Ahonen, T., Hadid, A., et al.: Face description with local binary patterns: Application to face recognition. PAMI (2006)
[7] Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled Faces in the Wild. TR UMass (2007)
[8] Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. PAMI (2002)
[9] yang, m.: kernel eigenfaces vs. kernel fisherfaces: face recognition using kernel methods. fg (2002)
[10] Vasilescu, M., Terzopoulos, D.: Multilinear image analysis for facial recognition. ICPR (2002)
[11] Zhao, W., Chellappa, R., Phillips, P., Rosenfeld, A.: Face recognition: A literature survey. ACM Computing Surveys (2003)
[12] He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.: Face recognition using laplacianfaces. PAMI (2005)
[13] Hua, G., Viola, P., Drucker, S.: Face recognition using discriminatively trained orthogonal rank one tensor projections. CVPR (2007)
[14] Hua, G., Akbarzadeh, A.: A robust elastic and partial matching metric for face recognition. ICCV (2009)
[15] Guillaumin, M., Verbeek, J., Schmid, C.: Is that you? Metric learning approaches for face identification. ICCV (2009)
[16] Wright, J., Hua, G.: Implicit elastic matching with random projections for pose-variant face recognition. CVPR (2009)
[17] Zou, J., Ji, Q., Nagy, G.: A Comparative Study of Local Matching Approach for Face Recognition. IEEE Transactions on Image Processing (2007)
[18] ORL Face Set: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html (Accessed 06/20/2010)
[19] Yale Face Set: http://cvc.yale.edu (Accessed 06/20/2010)
[20] CVL Face Set: http://www.lrv.fri.uni-lj.si/facedb.html (Accessed 06/20/2010)
[21] AR Face Set: http://cobweb.ecn.purdue.edu/ aleix/ar.html (Accessed 06/20/2010)
[22] Phillips, P., Moon, H., Rizvi, S., Rauss, P.: The FERET evaluation methodology for face-recognition algorithms. PAMI (2000)
[23] Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: Multi-PIE. Image and Vision Computing (2009)
[24] Pinto, N., DiCarlo, J.J., Cox, D.D.: Establishing Good Benchmarks and Baselines for Face Recognition. ECCV (2008)
[25] Pinto, N., DiCarlo, J.J., Cox, D.D.: How far can you get with a modern face recognition test set using only simple features? CVPR (2009)
[26] Taigman, Y., Wolf, L., Hassner, T., Tel-Aviv, I.: Multiple one-shots for utilizing class label information. BMVC (2009)
[27] Wolf, L., Hassner, T., Taigman, Y.: Similarity Scores based on Background Samples. ACCV (2009)
[28] Kumar, N., Berg, A.C., Belhumeur, P.N., Nayar, S.K.: Attribute and Simile Classifiers for Face Verification. ICCV (2009)
[29] Cao, Z., Yin, Q., Tang, X., Sun, J.: Face recognition with learning-based descriptor. CVPR (2010)
[30] Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol Cybernetics (1980)
[31] Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Robust object recognition with cortex-like mechanisms. PAMI (2007)
[32] Mutch, J., Lowe, D.G.: Object class recognition and localization using sparse features with limited receptive fields. IJCV (2008)
[33] Pinto, N., Cox, D.D., DiCarlo, J.J.: Why is Real-World Visual Object Recognition Hard. PLoS Comput Biol (2008)
[34] Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the best multi-stage architecture for object recognition? ICCV (2009)
[35] Pinto, N., Doukhan, D., DiCarlo, J.J., Cox, D.D.: A high-throughput screening approach to discovering good forms of biologically inspired visual representation. PLoS Comput Biol (2009)
[36] Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. CVPR (2004)
[37] Griffin, G., Holub, A., Perona, P.: The caltech-256 object category dataset. TR Caltech (2007)
[38] Gehler, P., Nowozin, S.: On feature combination for multiclass object classification. ICCV (2009)
[39] Geisler, W.S., Albrecht, D.G.: Cortical neurons: isolation of contrast gain control. Vision Research (1992)
[40] Rolls, E.T., Deco, G.: Computational neuroscience of vision. Oxford Univ Press (2002)
[41] Sonnenburg, S., Rätsch, G., Schäfer, C., Schölkopf, B.: Large Scale Multiple Kernel Learning. JMLR (2006)
[42] Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. CVPR (2006)