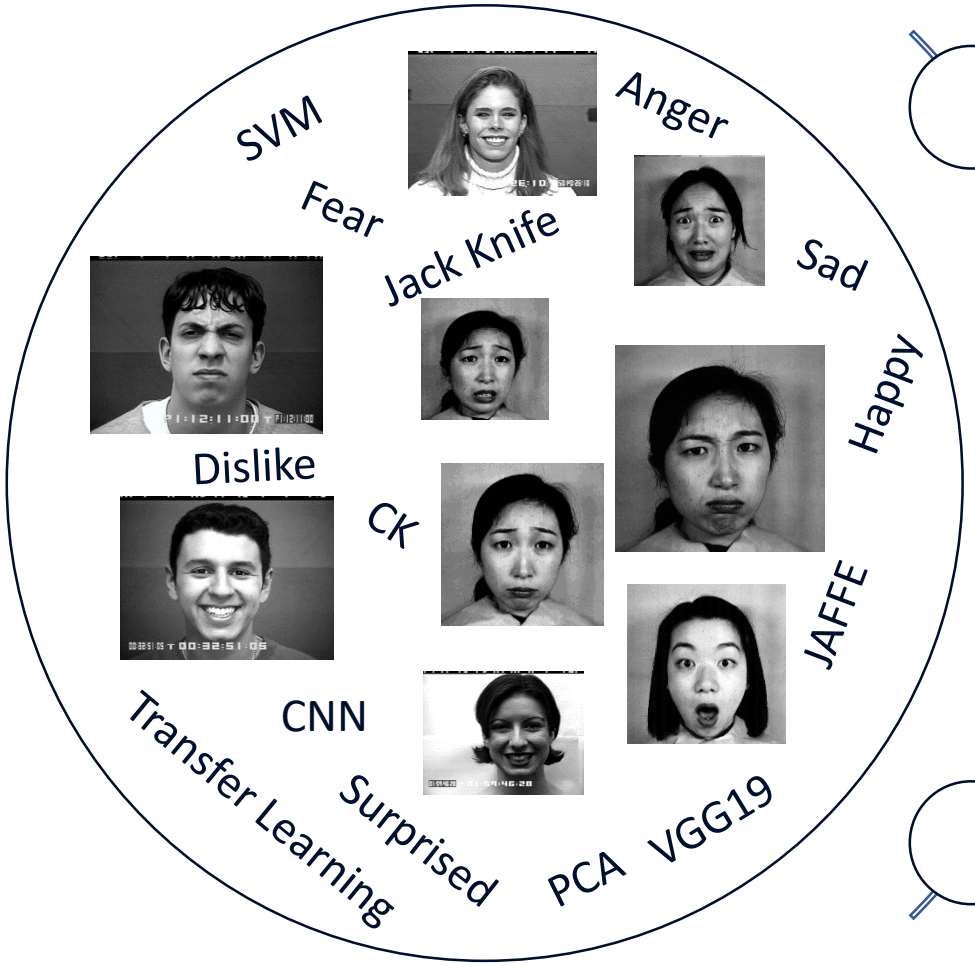
Pre-Trained Convolutional Neural Network Features for Facial Expression Recognition

Aravind Ravi
Department of Systems Design Engineering
University of Waterloo

Contents



Overview

Description

- A method for facial expression recognition based on transfer learning techniques is studied in this work
- The use of pre-trained convolutional networks (CNN) for feature extraction on a small dataset is presented
- This study was performed on the JAFFE dataset and a subset of CK+ dataset for features extracted from pre-trained CNN VGG19 (pre-trained on ImageNet Dataset)
- Principal Component Analysis (PCA) is applied for dimensionality reduction and feature selection
- A Linear Support Vector Machine (SVM) is used to classify the seven classes of facial expressions based on the selected features on both sets of data

Transfer Learning

"The ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks."

- *Broad Agency Announcement (BAA) 05-29 of Defense Advanced Research Projects Agency (DARPA)'s Information Processing Technology Office (IPTO)*

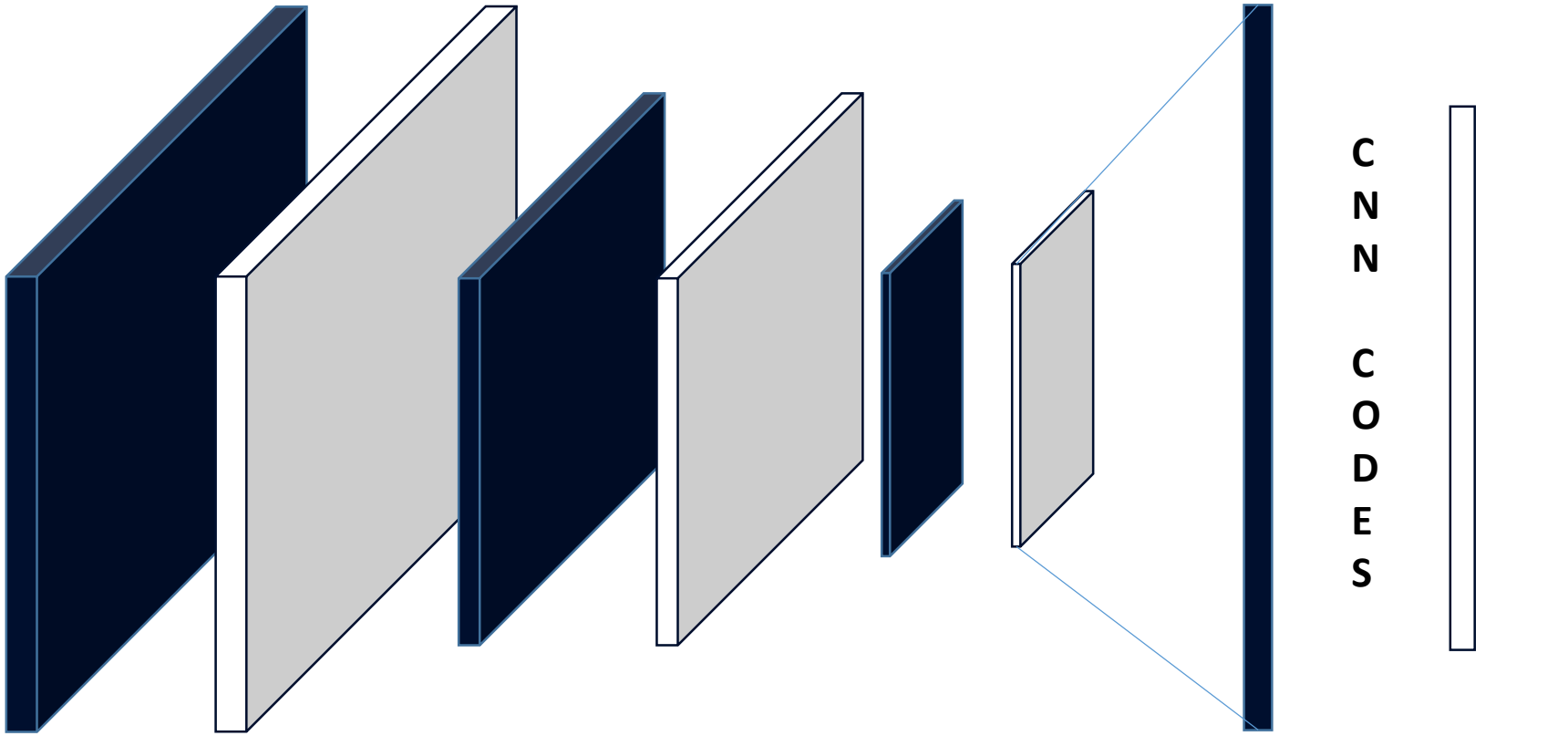
Transfer learning aims to extract the knowledge from one or more **source tasks** and applies the knowledge to a **target task**.

What is the objective?
To learn a "good" feature representation for the target domain

What is transferred?
The knowledge used to transfer across domains is encoded into the learned feature representation

References
Pan, Sinno Jialin, and Qiang Yang. "A survey on transfer learning." IEEE Transactions on knowledge and data engineering 22.10 (2010): 1345-1359.

Convolutional Neural Networks



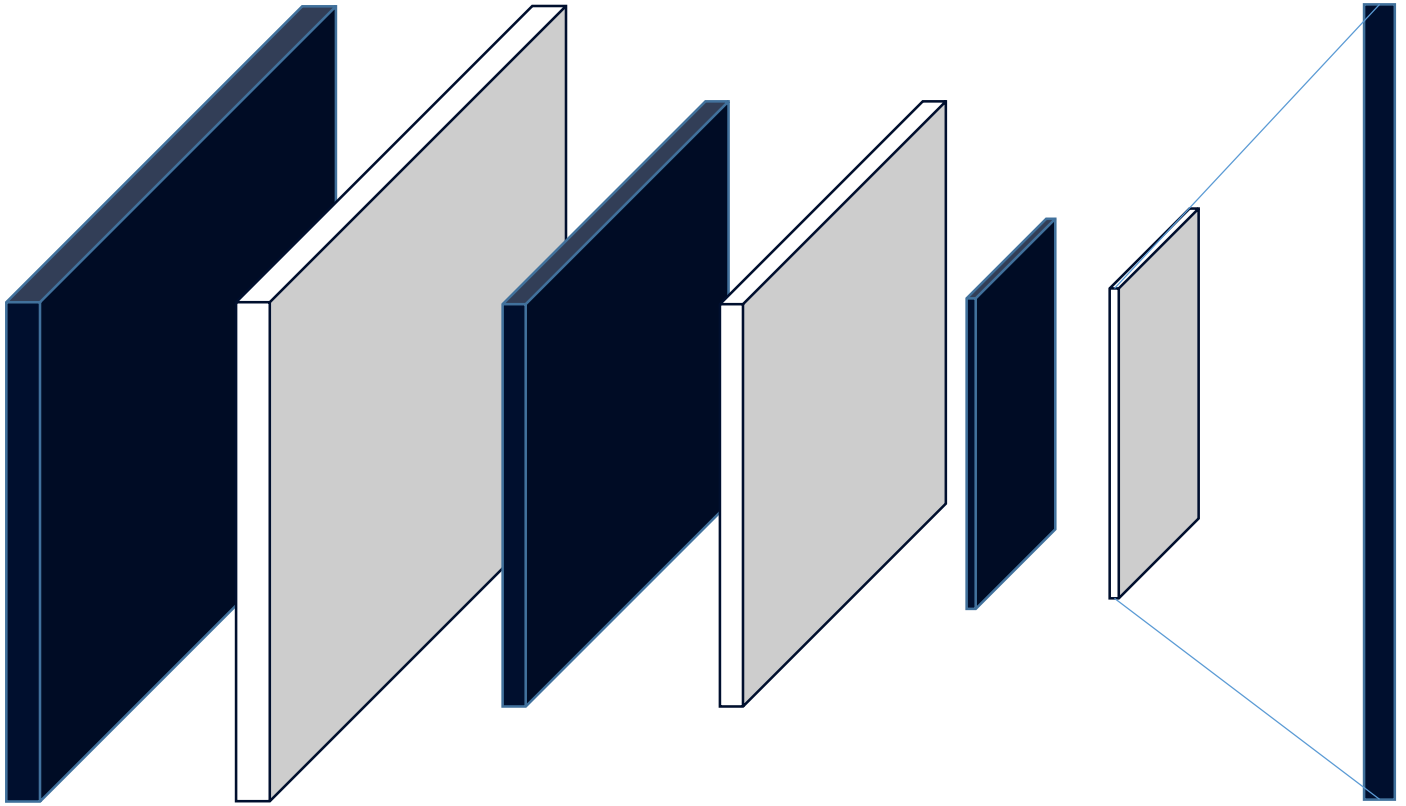
Conv Layer 1 | Pool Layer 1 | Conv Layer 2 | Pool Layer 2 | Conv Layer 3 | Pool Layer 3 | Fully Connected | Output Layer

Feature Extractor | **Classifier**

The earlier features of a CNN contain more generic features (e.g. edge detectors or colour blob detectors).

The later layers of the CNN becomes progressively more specific to the details of the classes contained in the original dataset.

Transfer Learning



CNN CODES

Linear Classifier (SVM)

Conv Layer 1 | Pool Layer 1 | Conv Layer 2 | Pool Layer 2 | Conv Layer 3 | Pool Layer 3 | Fully Connected

Feature Extractor

Classifier

Scenarios to Apply Transfer Learning

New dataset is small and similar to original dataset
Since the data is small, fine tuning the model could lead to overfitting. Due to similarity in the dataset, we expect higher-level features in the CNN to be relevant to this dataset as well. Hence, a linear classifier can be trained on the CNN codes.

New dataset is large and similar to the original dataset
Since the dataset is large, the model can be fine tuned.

New dataset is small but very different from the original dataset
Since the data is small, a linear classifier can be trained. Since the dataset is very different, the classifier can be trained from activations from somewhere earlier in the network, as deeper layers are more dataset specific.

New dataset is large and very different from the original dataset
Since the dataset is very large, the CNN can be trained from scratch. In practice, the CNN weights are initialized with weights from a pre-trained model.

References
http://cs231n.github.io/transfer-learning/#tf
Razavian, Ali Sharif, et al. "CNN features off-the-shelf: an astounding baseline for recognition." Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on. IEEE, 2014

Related Work

- Histopathology Images Using VGG16, Inception-V3

  Kieffer, Brady, et al. "Convolutional Neural Networks for Histopathology Image Classification: Training vs. Using Pre-Trained Networks." arXiv preprint arXiv:1710.05726 (2017).

- Medical Imaging for Lymph Node Detection and Interstitial Lung Disease Classification

  Shin, Hoo-Chang, et al. "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning." IEEE transactions on medical imaging 35.5 (2016): 1285-1298.

- Chest Pathology Detection Using Decaf pre-trained CNN model

  Bar, Yaniv, et al. "Chest pathology detection using deep learning with non-medical training." Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on. IEEE, 2015.

Basic Idea:
To use “off-the-shelf CNN” features (without retraining the CNN) as complementary information channels to existing hand-crafted image features

VGG19 - Architecture

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	**(None, 112, 112, 64)**	**0**
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	**(None, 56, 56, 128)**	**0**
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv4 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	**(None, 28, 28, 256)**	**0**
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv4(Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	**(None, 14, 14, 512)**	**0**
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv4(Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	**(None, 7, 7, 512)**	**0**
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	**(None, 4096)**	**102764544**
fc2 (Dense)	(None, 4096)	16781312
predictions (Dense)	(None, 1000)	4097000

Layers of Interest

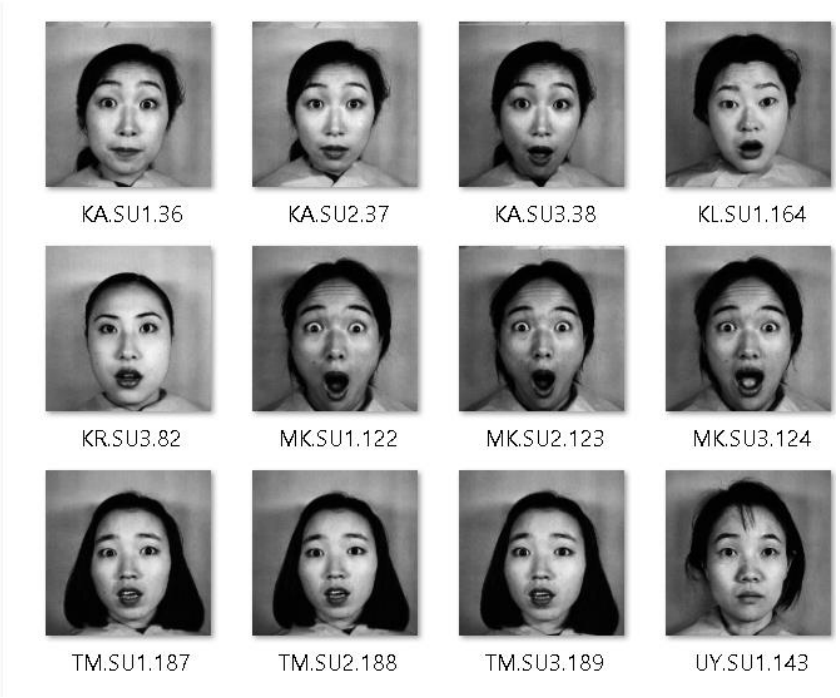
- block1_pool
- block2_pool
- block3_pool
- block4_pool
- block5_pool
- fc1

Total params: 143,667,240
Trainable params: 143,667,240
Non-trainable params: 0

References
Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014)

Dataset Description

Datasets



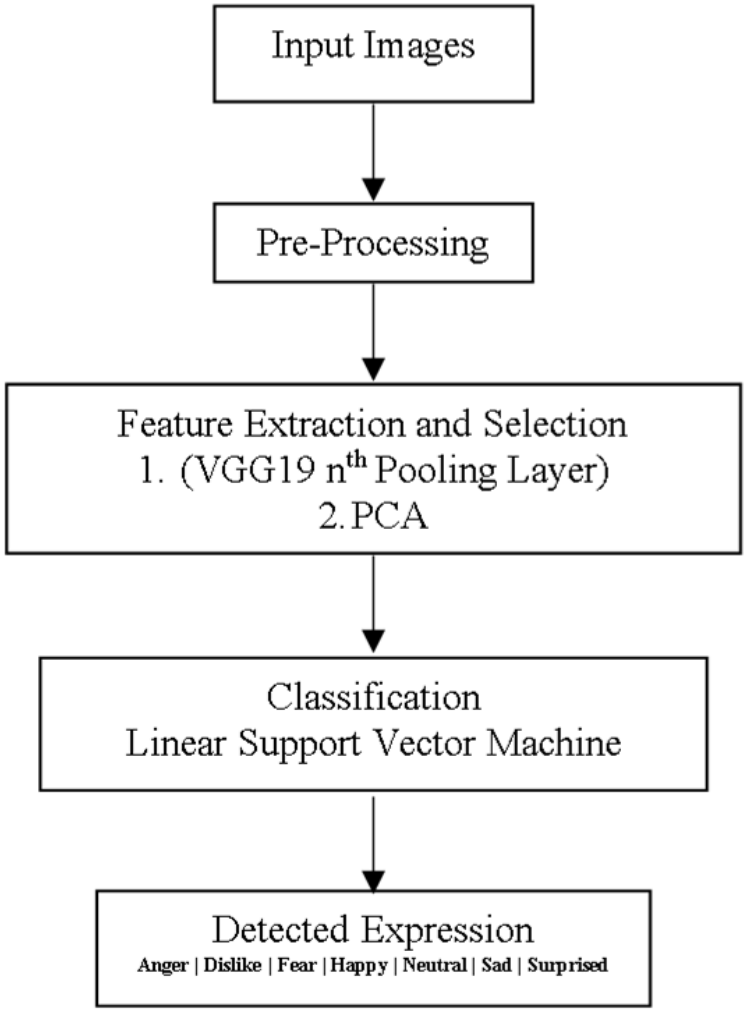
JAFFE – Japanese Female Facial Expression
Number of Subjects: 10
Number of Images: 213
Number of Expressions: 7

CK+ – Extended Cohn Kanade Dataset (Subset)
Number of Subjects: 10
Number of Images: 210
Number of Expressions: 7

References
Lyons, Michael, et al. "Coding facial expressions with gabor wavelets." Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on. IEEE, 1998.
Kanade, Takeo, Jeffrey F. Cohn, and Yingli Tian. "Comprehensive database for facial expression analysis." Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on. IEEE, 2000.

System Design

System Design



Methodology

Methodology

A. Feature Extraction

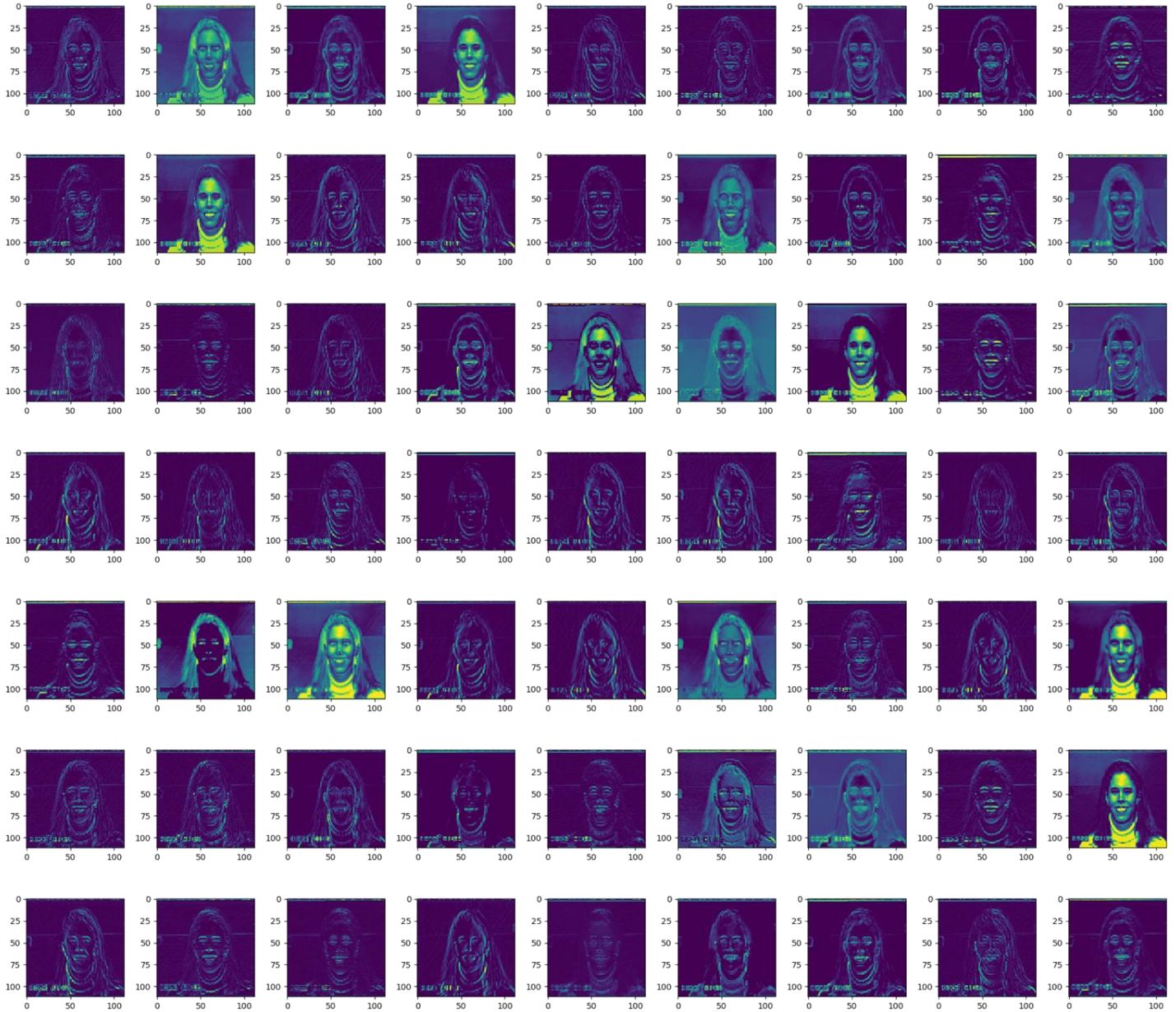
1. Features from each pooling layer and the first fully connected layer of VGG19 are extracted
   - ❖ Using the implementation provided for specific architectures within Keras
   - ❖ Each layer's output is vectorised and used as a feature vector

```
base_model = applications.vgg19.VGG19(include_top=True, weights='imagenet', input_tensor=None,
input_shape=None, pooling=None, classes=1000)

model = Model(input=base_model.input, output=base_model.get_layer('fc1').output)

features = model.predict(input_image)
```

References
F. Chollet et al., "Keras," https://github.com/fchollet/keras, 2015.

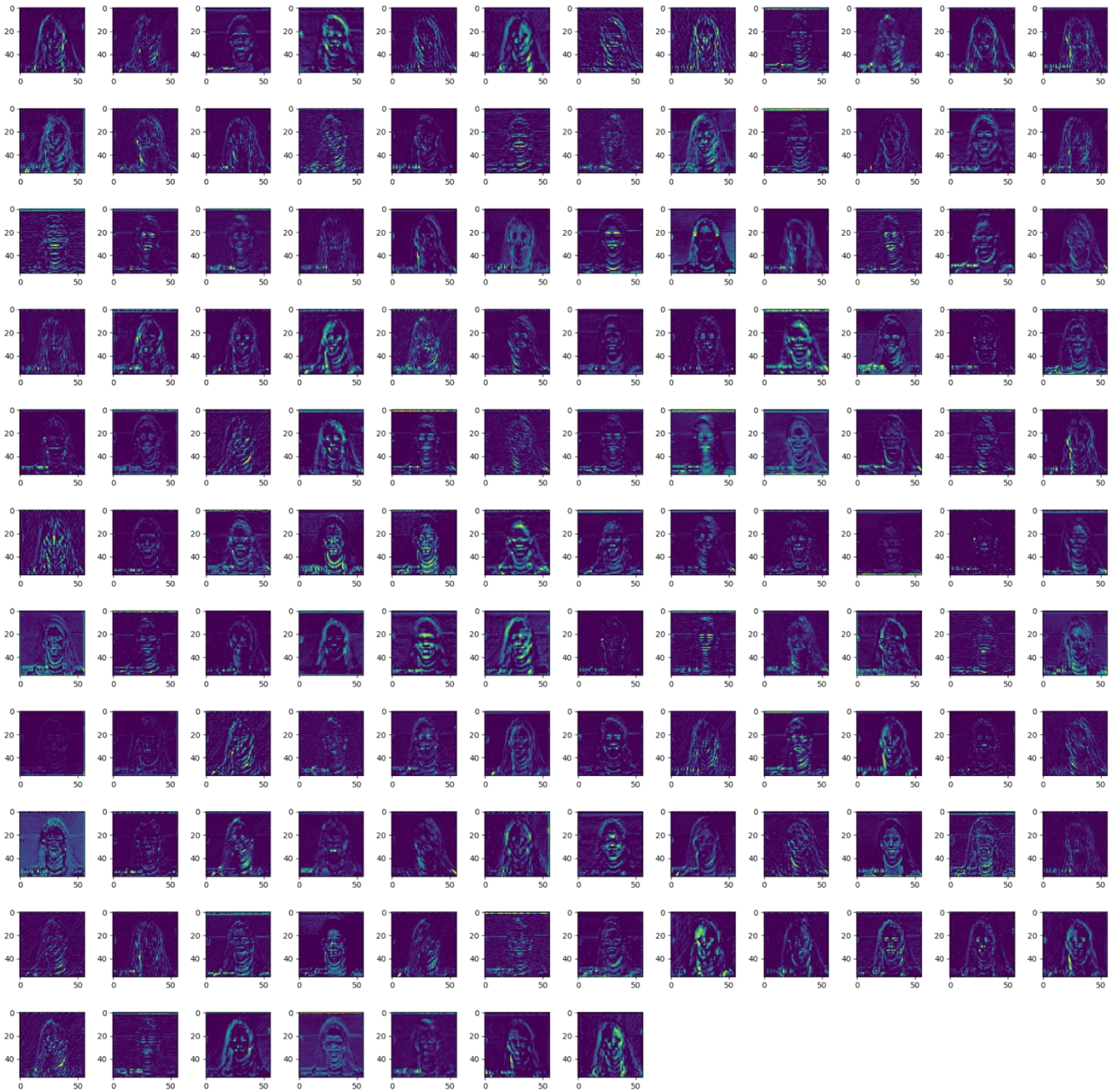
Block1_Pool

Size: (None, 112, 112, 64)



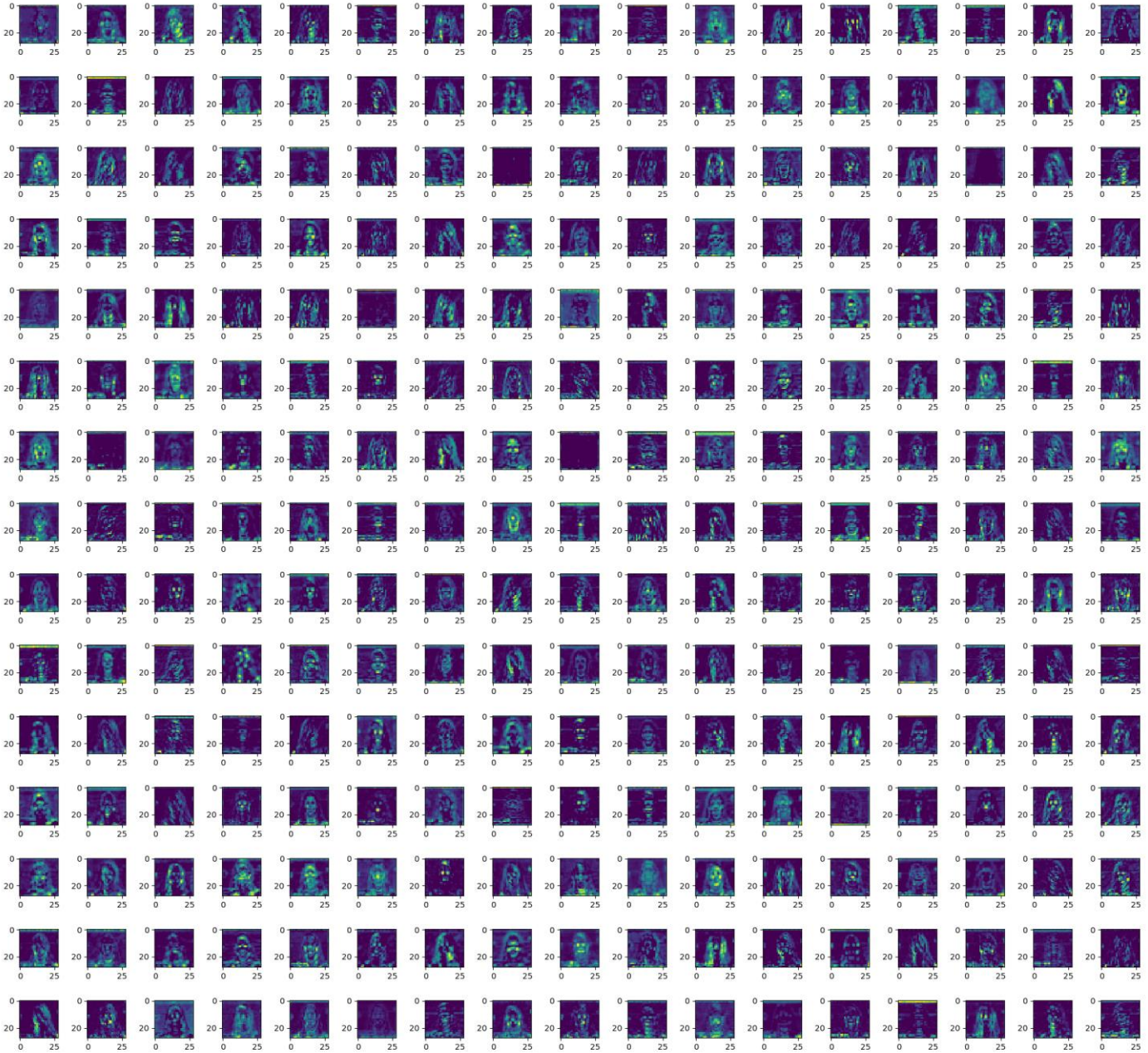
Block2_Pool

Size: (None, 56, 56, 128)



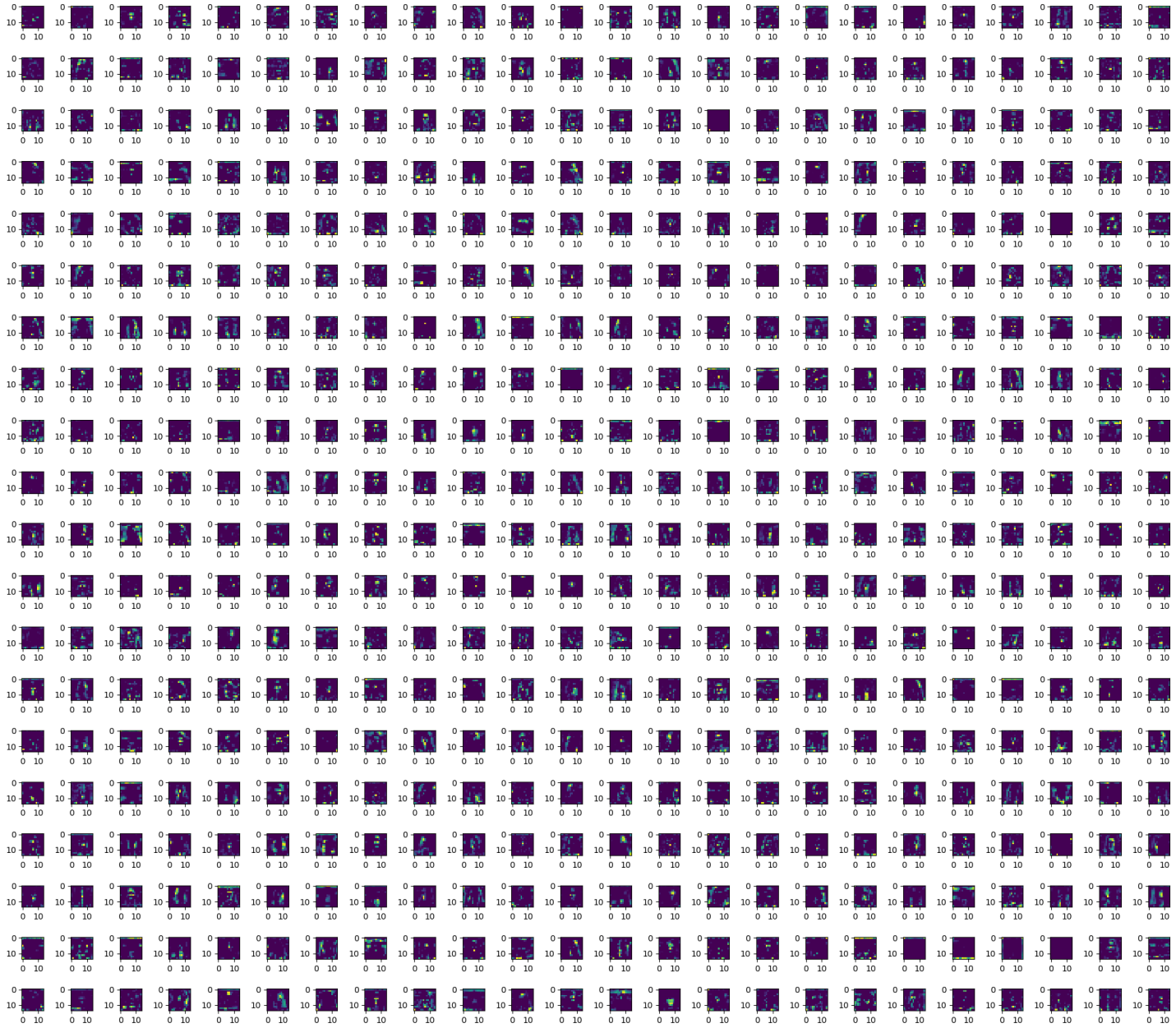
Block3_Pool

Size: (None, 28, 28, 256)



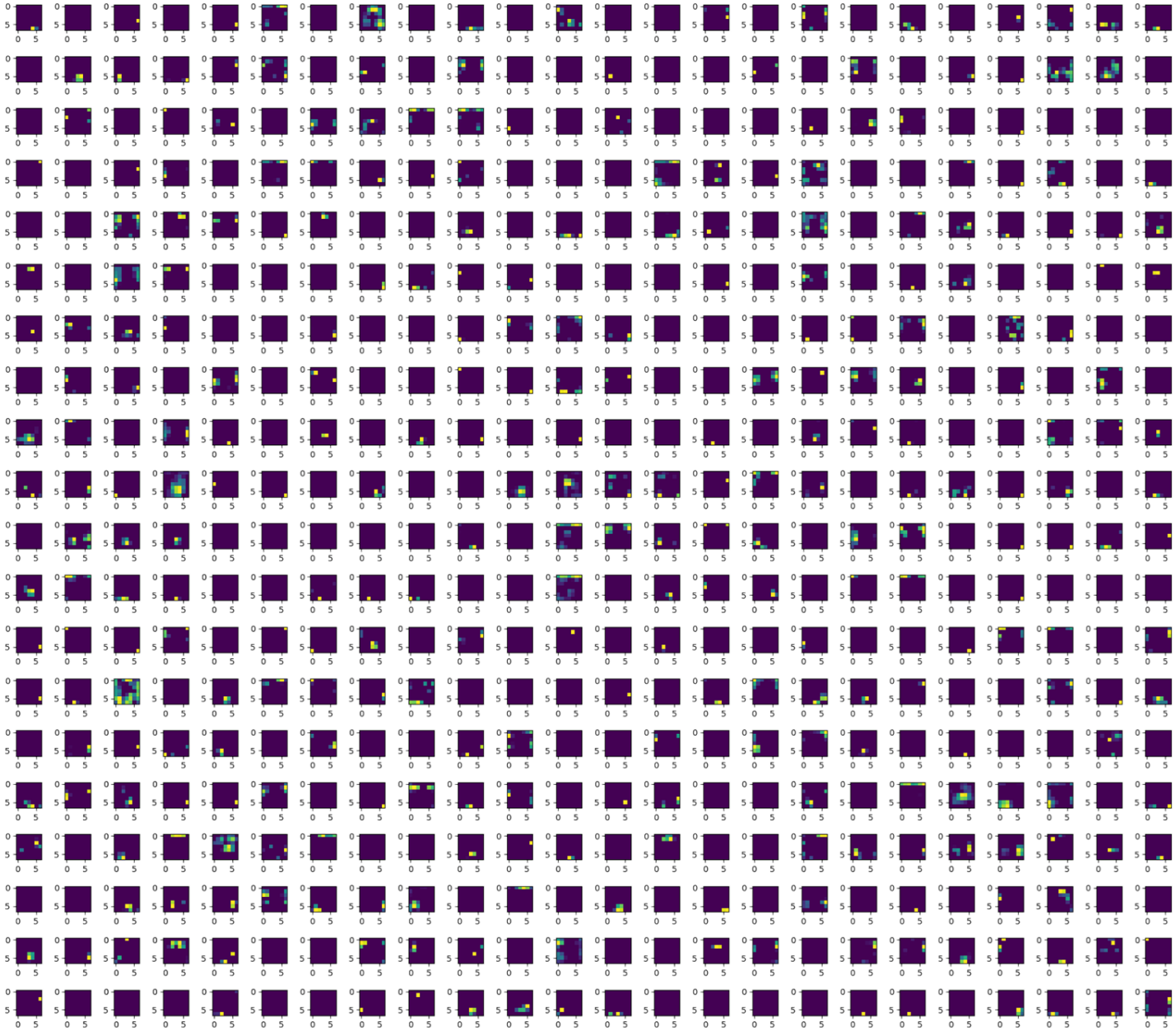
Block4_Pool

Size: (None, 14, 14, 512)



Block5_Pool

Size: (None, 7, 7, 512)



Methodology

A. Feature Extraction (Continued)

2. Reduce dimensionality based on chosen number of PCA components (N_{PCA})
3. Four values for the choice of components were evaluated {50,100,150,200}

B. Feature Selection

This is based on two parameters:

1. Nth VGG19 Layer for feature extraction
2. Find N_{PCA} for dimensionality reduction and feature selection

Method:

1. Select the parameters of the two highest accuracies from the Jack-Knife validation results.
2. Select the final parameters based on the configuration with the least difference between the Jack-Knife training accuracy (A_{JK}) and the Test accuracy (A_T)

Feature Vector Size (Per Layer)
Block1_Pool – 802,816
Block2_Pool – 401,408
Block3_Pool – 200,704
Block4_Pool – 100,352
Block5_Pool – 25,088
FC1 - 4096

Methodology

C. Classification

- A Support Vector Machine with a linear kernel is used as the classifier
- The Python package scikit-learn was used to train the SVM and NumPy was used to process and store the data during the experiments

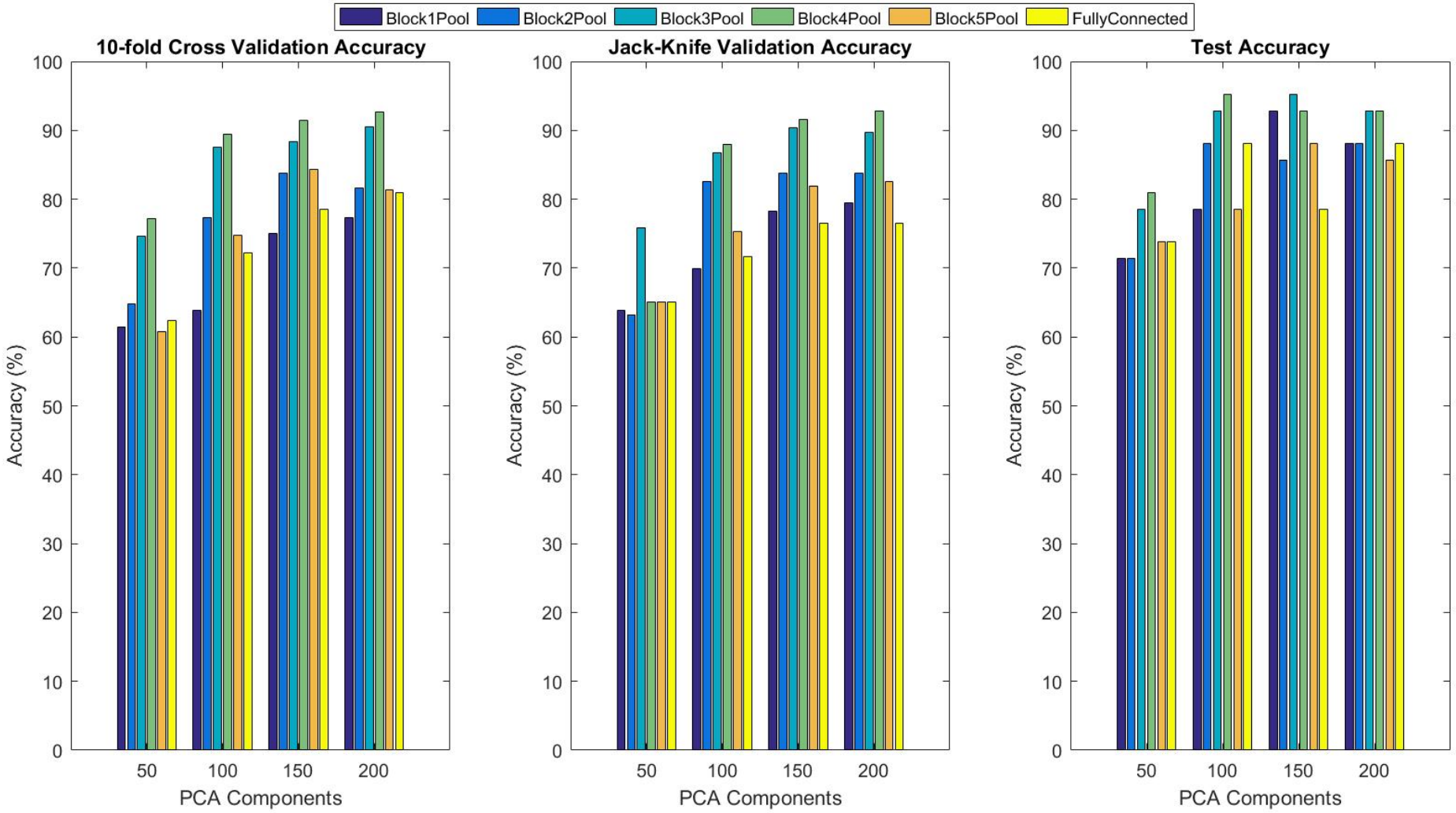
D. Validation

- 80% of the training data was used to train the classifier and 20% was used for testing
- These results are validated based on
    1. 10-fold cross-validation
    2. Leave-One-Out validation/Jack-Knife method (Due to the small size of the dataset)

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

Results and Discussions

Results - JAFFE



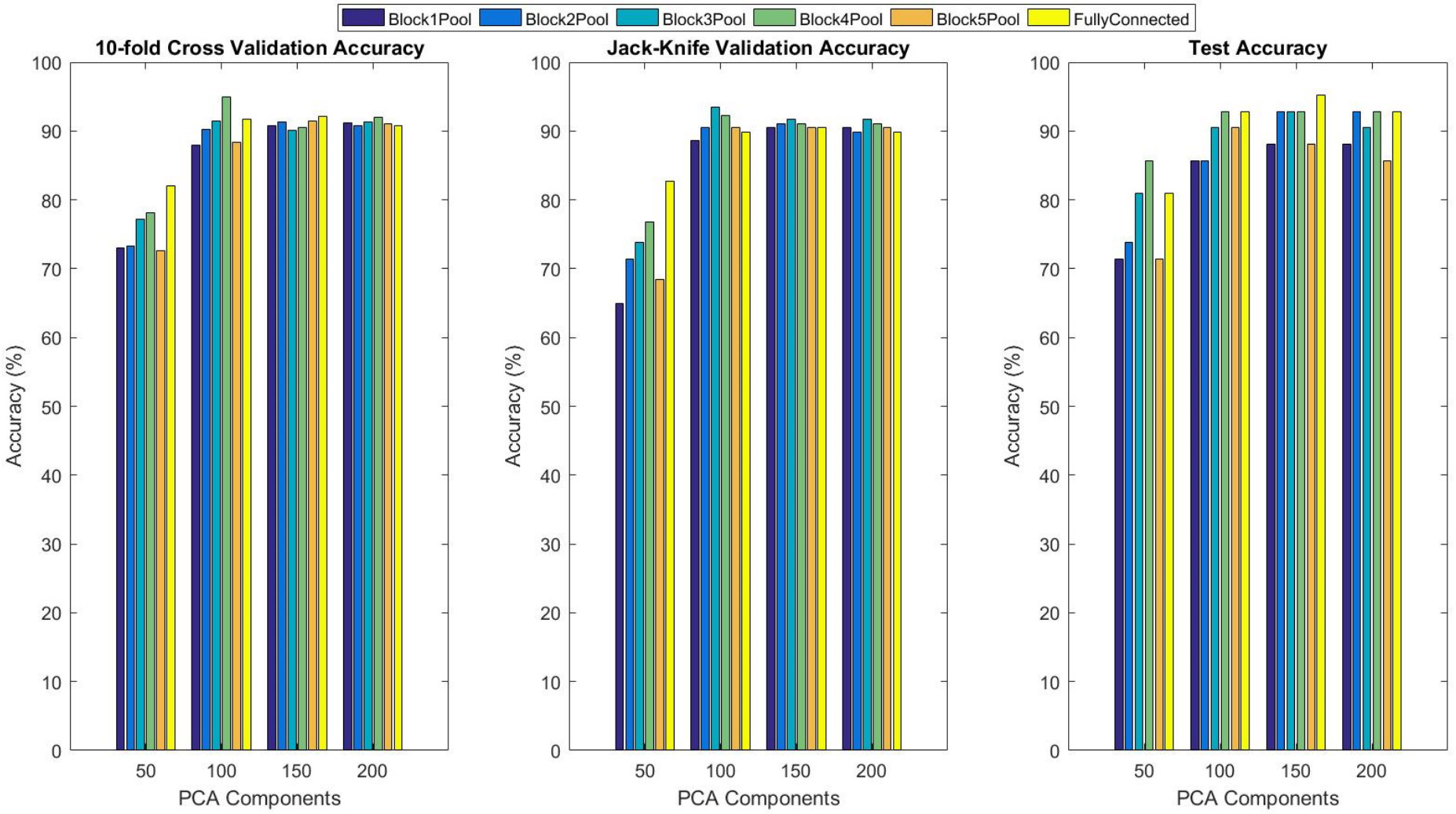
Selected Features

- 200 PCA Components
- Block4_Pool Layer

Results

- 10-Fold Acc. = 92.70 %
- A_{JK} = 92.77%
- A_T = 92.86 %

Results - CK



Selected Features

- 100 PCA Components
- Block4_Pool Layer

Results

- 10-Fold Acc. = 94.93 %
- A_{JK} = 92.26%
- A_T = 92.86 %

Results

	CK+ Dataset			JAFFE Dataset		
Features	PCA - 100			PCA - 200		
CNN Layer (Pool)	Training Data (80%)		Test Data (20%)	Training Data (80%)		Test Data (20%)
	10-Fold	Jack-Knife (A_{JK})		10-Fold	Jack-Knife (A_{JK})	
block1	87.90	88.69	85.71	77.27	79.52	88.10
Block2	90.27	90.48	85.71	81.66	83.73	88.10
Block3	91.51	93.45	90.48	90.47	89.76	92.86
Block4	**94.93**	**92.26**	**92.86**	**92.70**	**92.77**	**92.86**
block5	88.83	90.48	90.48	81.35	82.53	85.71
(Dense) FC1	91.76	89.88	92.86	81.01	76.51	88.10

:

Discussions

- Among the different methods of feature extraction, features from **block4_pool** layer of VGG19 provides the highest accuracy for both CK+ and JAFFE dataset.
- Applying the proposed feature selection methods (PCA components), a training accuracy of **92.77%** and **92.86%** accuracy on test set was achieved for the **JAFFE dataset**.
- A training accuracy of **92.26%** and test accuracy of **92.86%** was achieved on the **subset of CK+ dataset**.
- These results suggest that representations learnt from pre-trained networks trained for a particular task such as object detection can be transferred, and used for a different task such as facial expression recognition.
- Furthermore, for a small dataset, using features from earlier layers of the network provide better accuracy.

:

Q&A

Thank You

Aravind Ravi
Department of Systems Design Engineering
University of Waterloo
aravind.ravi@uwaterloo.ca

: