

Received January 5, 2019, accepted January 15, 2019, date of publication January 21, 2019, date of current version February 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2894006

Feature Learning and Analysis for Cleanliness Classification in Restrooms

LAHIRU JAYASINGHE¹, NIPUN WIJERATHNE¹, CHAU YUEN¹, (Senior Member, IEEE),
AND MENG ZHANG²

¹SUTD-MIT International Design Centre, Singapore University of Technology and Design, Singapore 487372

²National ASIC System Engineering Technology Research Center, Southeast University, Nanjing 210096, China

Corresponding author: Lahiru Jayasinghe (aruna_jayasinghe@sutd.edu.sg)

This work was supported in part by SUTD-MIT International Design Center and NSFC 61750110529.

ABSTRACT In order to revamp the cleaning contract from the head-count basis into a performance basis, a fair and unbiased cleanliness classification is necessary. However, the perception of cleanliness is very subjective to the observer. Hence, it is not an easy task to quantify the cleanliness. This paper presents an application of principal component analysis (PCA) in conjunction with convolutional neural networks (CNN) to identify the cleanliness of a restroom up to three levels; namely, dirty, average, and clean. The proposed method includes an application specific data augmentation algorithm and a PCA-based feature analysis schema to select the best suited CNN model for our dataset. Since this study focused on a specific application, we benchmark the performances of the proposed method performances with the state-of-the-art computer vision algorithms on our dataset. Moreover, our study shows a machine learning approach toward automating the inspection process of a restroom.

INDEX TERMS Image classification, deep learning, principle component analysis, data augmentation, feature learning, internet of things.

I. INTRODUCTION

With the increasing of man power cost, there is a suggestion to move away from the traditional “head-count” system to a “performance-based” system for the cleaning services, i.e. instead of hiring a fixed number of cleaners, the cleaning services is charged based on the cleanliness performance. Hence, there is a need of cleanliness measuring standards, which provide precise definitions. One of such standard can be found in [1].

However, the interpretation of a cleanliness standard and the perception of cleanliness could still be subjective, and different persons could rank differently. A fully automated system that can provide measurement on the cleanliness according to the cleaning standard, would move away the human bias or time-consuming human training. Hence, the system should be equipped with Artificial Intelligence (AI) to make an autonomous decision by providing a fair judgment about the cleanliness based on certain cleaning standard. With such system, the building managers and cleaning service providers can then focus on performance-based cleaning system, while reducing the workload and dispute relevant to restroom cleaning process.

In the literature, there have been a number of work related to the restroom cleanliness. A research group from [2] has

proposed a smart cleaning solution called Restroom Visualizer System, which utilize people counters and odour sensors to determine the ammonia level present in restrooms. Two other research groups [3], [4] have proposed similar solutions, which utilize people counters and ammonia sensors to detect the dirtiness in restrooms. They have proposed an end-to-end solution for the facility managers to monitor the cleanliness in their restrooms.

To this end, the previous work focused on the odour based sensor systems to detect the dirtiness. However, there could be other odourless dirtiness that can appear in a restroom. For instance, hand touching places like taps, doorknobs, sink mirrors easily get dirty and tissue papers might be often spread all over the restroom. Odour based sensor systems need a considerable number of sensors to cover the whole restroom, which can be sometimes inconvenient. Furthermore, according to the [3], limited interpretability of existing odour sensors and analyzing complex correlations among readings from various sensors deployed in different facilities are challenges in existing systems.

In this paper, we aim to design an intelligent architecture, that can access the restroom cleanliness based on certain standards [1]. We present a vision based approach to classify the odourless dirtiness in restrooms by employing PCA for



FIGURE 1. Data samples from each category in dataset \mathcal{S} shown here. Images in (a), (b), and (c) represent data relevant to *dirty*, *average*, and *clean* categories respectively.

feature analysis, CNN for feature extraction, and a neural network for classification. In this work, we focused on analyzing more effective features with respect to the dirtiness level, rather than analyzing the functional structure or the shape of object as in many object classification methods [5]. Moreover, our focus is not just classifying objects based on the shape or the geometry, but understand the deep underlying features with respect to cleanliness that will eventually guide the classification stage. For this purpose, we comprises images of urinal bowls that are captured by a normal smart phone camera. The data labeling is provided by professional building managers, where they have labeled the images into three categories, namely *clean*, *average* and *dirty*.

The contributions of our work can be summarize as follows. There are imbalanced data across different categories, specially in the dirty case that is impossible to cover all potential dirty scenarios. In order to tackle this problem, we introduced a PCA based color augmentation method to enhance the dirtiness in images or to introduce new dirt in to the image. Next, to perform classification, the usual procedure for the transfer learning is to use several pre-trained CNN models like VGG-16, ResNet-50 or Inception-V3 and fine-tune or retrain those one by one, and at the end compare the classification accuracies for each trained models. But this process consume lot of computational resources and its very time inefficient. We introduced a robust PCA analysis schema to skip this above mentioned process to some extend and

select the most suitable CNN feature extractor for the data without undergoing any retraining or fine-tuning process. Then extracted features from selected feature extractor are classified using a neural network classifier. Finally, even though the solution need not to be generalized into other restrooms, we still performed a brief experiment using our trained model with some urinal bowl images taken from other restrooms. The results from this experiment showed that our proposed model can still accurately perform the classification up to a acceptable level in other restrooms. However, such generalization deserves in-depth study, which will be treated as a future work.

Rest of the paper is organized as follows. State-of-the-art image feature extraction methods and classification methods are discussed in Section II. The proposed method is presented in Section III and their results are discussed in Sections IV separately. Finally, the Section V concludes the paper.

II. LITERATURE REVIEW

State-of-the-art techniques for image feature learning and how these learned features have being used in the image classification are reviewed in this section. In particular, we present three recent DCNN architectures that have been proposed for image classification.

A. IMAGE FEATURE LEARNING

The most significant features of an image can be stated as its color and the texture. Usually, color information

included of an image could be measured by color histograms (RGB, HSV and LAB histograms), while texture details are described using LBP histograms and Gabor filters [6]. One famous method for image feature analysis is Bag-of-feature (BoF) [7]. It contains three phases, namely; feature extraction, feature encoding, and feature pooling. The feature extractor uses SIFT [8], HOG [9], or SURF [10] techniques to extract features, while encoding is performed by sparse coding related methods such as [11] and [12], and BoF feature pooling could be performed using Spatial Pyramid Matching (SPM) or max-pooling [13], [14]. Moreover, literature describe that the PCA can be used to learn robust features in an image. In [15] outlined a method of combining PCA and ICA (Independent Component Analysis) for face recognition. Ke and Sukthankar [16] have implemented distinctive descriptors using PCA and SIFT.

Recently, there has been much research conducted on feature learning using Convolutional Neural Networks [17]. With the recent development in CNN, it has been commonly applied in diverse real-world applications such as medical image analysis [18], time series analysis [19], speech recognition [20], etc. Generally, CNN occupies three main layers; convolutional layer, pooling layer, and fully connected layer. DCNN contains a large number of deep stacks of these layers, hence the name DCNN. It has the ability to learn meaningful features from a given dataset, while iteratively minimizing the error. Intuitively, when the number of layers grow, the performances will also improve along with it. When DCNN gets deeper, the parameter count will also increase exponentially and those parameters could be able to explain almost every important feature about the dataset. However, to extract meaning full features, DCNN needs a vast amount of sample data. The ImageNet dataset can be recognize as a large-scale dataset for image recognition [21]. Many DCNN architectures were introduced for image classification and detection using ImageNet's Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) dataset. AlexNet [22], VGG [23], ResNet [24] and Inception-V3 [25] are some notable DCNN architectures trained using ImageNet dataset.

B. IMAGE CLASSIFICATION

The most efficient method of image classification is to extract features of the images and classify them based on their features using a particular classifier. This pipeline is used in BoF and DCNN based image classifications.

BoF depicts the image as histograms of discrete quantized features and uses SVM as a classifier to perform the classification. Spatial pyramid approach introduced by Lazebnik *et al.* [13], which count the features inside a sub-region without focusing the image as a whole. Lin *et al.* [26] improved this classification approach by incorporating sparse coding and dictionary learning. This proposed implementation achieved the best performances in ILSVRC 2010 classification challenge. Thereafter, Perronnin *et al.* [27] employed fisher kernel to introduce higher order statistics for image classification and their approach achieved the

state-of-the-art image classification results in ILSVRC 2011. These higher order statistics can be considered as the baseline idea for deep learning.

After ILSVRC 2011, higher order statistical approaches for image classification got attention. As a result, AlexNet was introduced [22]. AlexNet consists of five convolutional layers followed by two fully connected and a softmax output layer. This architecture achieved the top-5 error rate of 15.3% and won the ILSVRC 2012. As mentioned before, the same pipeline approach is used for AlexNet, which means extracting the features by employing a DCNN and classify them using a soft-max classifier. From this point onwards, all the winning architectures were based on DCNNs with various modifications. The Spatial Pyramid Pooling (SPP-net [28]), batch-normalized networks, inception modules, and residual networks can be noted as some iconic modifications in DCNN architectures. Some of the state-of-the-art DCNN architectures are described below.

1) VGG-16

The VGG network architecture was introduced by [23] for the ILSVRC 2014 challenge. This architecture uses only 3×3 convolutional layers stacked together to increase the depth, while reducing the volume size by using max pooling. Two fully-connected layers, each with 4,096 nodes are then followed by a softmax classifier which includes 1000 classes, see Fig. 2a. This architecture demonstrates astonishing results in 2014, and it is frequently being used in many deep learning applications.

2) RESNET-50

Sometimes deep architectures performs worse when they become deeper. He *et al.* [24] have proposed a solution for this issue by introducing residual networks into DCNNs, see Fig. 2b. Intuitively, if a DCNN is capable of extracting meaningful features, then its accuracy should not be degraded by adding one layer of network, this is because the last layer of the original network should be able to learn an identity feature mapping to the newly added layer. However, in practice, deep neural nets suffer from a vanishing gradient problem, which means the gradient signal could go to zero quickly when back propagating through layers, while making a gradient descent unbearably slow or impossible. Bringing in the residual models, the gradient signal could travel backward via skip connections in residual nets. Therefore, one can suddenly build 50-layer or even 1,000 layer nets using residual blocks.

3) INCEPTION-V3

Though the ResNet is about going deeper, the Inception architecture is all about going wider. Inception-V3 is a very deep network whose architecture is inspired by Network-in-Network concept [29] and sparse networks. The architecture as shown in Fig. 2c consists of an image stream, eleven inception modules, and a fully connected neural network classifier. Inception-V3 focuses on developing a deeper network without increasing much computational cost. The whole architecture contains 54 neural layers with 25 million parameters.

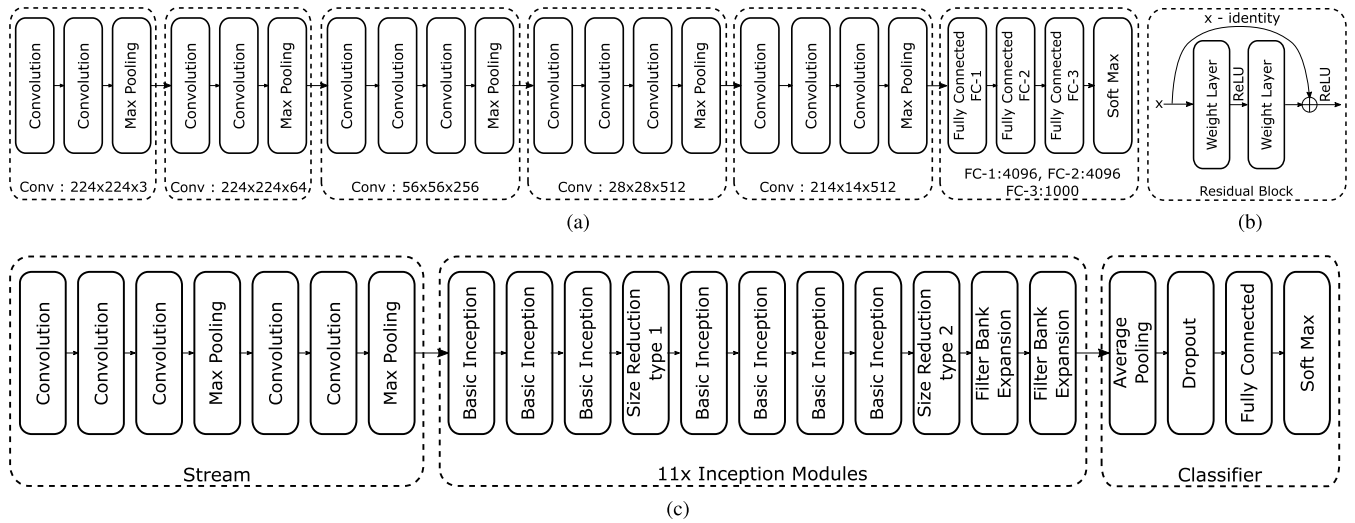


FIGURE 2. State-of-the-art DCNN architectures and their building blocks are shown here. (a) VGG-16 DCNN architecture, (b) the fundamental building block of ResNet-50 DCNN architecture and (c) is the Inception-V3 DCNN architecture.

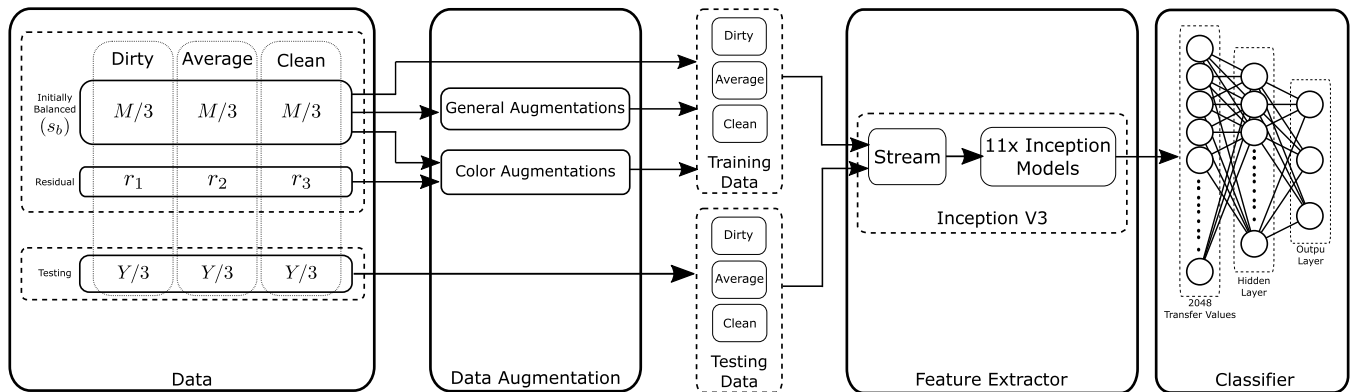


FIGURE 3. The system pipeline of the proposed methodology.

The input image stream consists of five convolutional layers along with two max-pooling layers. The most crucial part of feature extraction in Inception-V3 model is the stack of eleven inception modules. Inception stack is made of four types of different inception modules, namely, basic inception module, size reduction module, and two types of filter bank expansion modules. The basic inception module is designed to approximate the optimal local sparse structure of features. Size reduction modules are responsible for reducing model dimension otherwise the computational requirements would be too complex. The two types of filter bank expansion modules assist to increase the dimensional representations for the soft-max classifier.

III. PROPOSED METHODOLOGY

As shown in Fig 3, the proposed methodology consists of data augmentation, feature extraction, and a classifier. The complete dataset \mathcal{S} consists of 4032×3024 pixels sized RGB urinal bowl images taken from a restroom facility. As shown in Fig. 1, the dataset \mathcal{S} is labeled by an experience building manager into three classes, namely: dirty, average, and clean.

Since urinal bowls not frequently getting dirty, the number of images in dirty and average categories are less than the number of images in the clean category. He and Garcia [30] investigated that the algorithms trained with balanced datasets usually surpass those trained with imbalanced. Hence, we prepared an balanced dataset S_b of size M by removing r_1 , r_2 , and r_3 number of images from each category in dataset \mathcal{S} , and kept the these r_1 , r_2 , and r_3 number of images as residuals to accommodate in color augmentation and validation tasks (explained in subsection III-A and III-C). For the testing phase, we acquired a new Y number of images from the same restroom facility environment to ensure that there were no overlaps between training and testing data. Thereafter, a pre-trained Inception-V3 feature extractor was selected through a PCA analysis process. Finally, the extracted features are classified by using a single hidden layer neural network classifier.

A. DATA AUGMENTATION

Though there are many techniques to augment the data, a mere and plausible way could be random crops. Since the

dataset consists of rather high-resolution images, the cropped frame needs to incorporate a considerable part of the urinal bowl. Therefore, we included 50% and 75% crops from the original image and resize it to match with the input size of the feature extractor.

Images in the dataset contain variety of inconsistencies due to variations of camera angles and lighting conditions, see Fig. 1. For instance, images show various scales of urinal bowl sizes, some urinal bowls are flipped compared to others and lighting conditions are fairly inconsistent between data.

Furthermore, these disparities could influence the decision-making process. Subsequently, the classifier will learn these differences as the features for classification rather than learning the dirtiness feature included in the data, this phenomena is known as over-fit in deep learning. Thus, the solution would be to eliminate these dissimilarities by introducing additional data with the same types of disparities, then the classifier can be trained to be robust to variants during implementation. Therefore, we apply random scaling, horizontal flipping, and histogram equalizations. These techniques are referred as general augmentations in Fig 3.

The other form of data augmentation consists of altering the intensities of RGB channels probably around the dirty regions (outliers). The motivation is to increase the dirtiness by enhancing the pixels around the dirty region, see Table. 1.

Algorithm 1 Color Augmentation

Input: Complete dataset, $\mathcal{S} = \{S^n | n = 1, \dots, N\}$

Input: A image, S_b^n , from the balanced dataset

$$S_b = \{S_b^n | n = 1, \dots, M\}, S_b \subset \mathcal{S}$$

Output: Im_1, Im_2

Initialization: $r_{curr}, g_{curr}, b_{curr} \rightarrow \emptyset$;

for each image $n = 1, \dots, N$ **do**

$S' = f_{resize}(S^n)$;

Decomposition: $S' = [S'_{red}, S'_{green}, S'_{blue}]$;

$r_{curr} = f_{Concatenate}(vec(S'_{red}), r_{curr})$;

$g_{curr} = f_{Concatenate}(vec(S'_{green}), g_{curr})$;

$b_{curr} = f_{Concatenate}(vec(S'_{blue}), b_{curr})$;

end

Mean Centering:

$$I_{vec} = [r_{curr} - \bar{r}_{curr}, g_{curr} - \bar{g}_{curr}, b_{curr} - \bar{b}_{curr}];$$

PCA: $[\lambda, u] = PCA(I_{vec})$;

Eigenvectors: $u = [\bar{u}_1, \bar{u}_2, \bar{u}_3]$;

Eigenvalues: $\lambda = [\lambda_1, \lambda_2, \lambda_3]$;

$$\lambda_s = [\alpha_1 \lambda_1, \alpha_2 \lambda_2, \alpha_3 \lambda_3] \quad \forall \alpha_i \sim \mathcal{N}(\mu, \sigma^2);$$

$$v = u \lambda_s^T;$$

$$I'_{vec} = f_{Normalization}(S_b^n) + v;$$

$$Im_1 = S_b^n + \beta I'_{vec} \quad \forall \beta \sim \mathcal{N}(\mu, \sigma^2);$$

$$I'_{vec} = f_{Normalization}(I'_{vec});$$

$$I_{BW} = \gamma [f_{BinaryThresholding}(\beta I'_{vec})], \{\gamma \in \mathbb{R} | 0 < \gamma < 1\};$$

$$Im_2 = f_{Normalization}(I_{vec}) - I_{BW}$$

Result: Im_1, Im_2

The Algorithm 1 describes the color augmentation methodology. The first input is the complete dataset $\mathcal{S} = \{S^n | n = 1, \dots, N\} \forall S^n \in \mathbb{R}^{W \times H \times 3}$, where N is the size of complete

dataset and W, H are scalars indicating the height and width of a single RGB image. The second input is a image S_b^n from the balanced dataset $S_b = \{S_b^n | n = 1, \dots, M\}$, where M represents the number of data in S_b where $S_b \subset \mathcal{S}$. Due to the high-resolution (W and H) of images, a resizing operation performed using first order spline interpolation that can be denoted as,

$$S' = f_{resize}(S^n). \quad (1)$$

This helps to increase the computational efficiency by reducing the image dimensions. Since S' is a RGB image, its easy to decompose and represent it by color channels as $S' = [S'_{red}, S'_{green}, S'_{blue}]$, where S'_{red} , S'_{green} , and S'_{blue} represent the respective RGB color channels of the image S' . Then, vectorization of a color channel of the image S' can be denoted as $vec(S'_{color_channel})$, where $color_channel \in \{red, green, blue\}$.

According to Algorithm 1, vectorization applied to every image in the complete dataset \mathcal{S} and these vectorized color channels are concatenated along their respective color channel. The concatenation operation is denoted by $f_{concatenate}(\cdot, \cdot)$. The motivation of this adjustment is to obtain eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and eigenvectors $\bar{u}_1, \bar{u}_2, \bar{u}_3$ that describes details about the feature distribution of the complete dataset \mathcal{S} , and then use these values in the color augmentation.

After the loop mentioned in Algorithm 1, created the I_{vec} by taking $r_{curr} - \bar{r}_{curr}$, $g_{curr} - \bar{g}_{curr}$, and $b_{curr} - \bar{b}_{curr}$ as columns of the matrix I_{vec} , where \bar{r}_{curr} , \bar{g}_{curr} , and \bar{b}_{curr} represent the mean values of the r_{curr} , g_{curr} , and b_{curr} respectively. Thereafter, performed the PCA on I_{vec} and obtained the respective eigenvectors u and eigenvalues λ .

Then sampled a α_i scaler from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, where $i = \{1, 2, 3\}$, $\mu = 0$ and $\sigma = 0.15$. The values for μ and σ were empirically decided based on the dataset images. Then it was straight forward to calculate v , while calculating the λ_s by using sampled α_i 's [22]. Thereafter, we performed a channel wise image normalization to the input image S_b^n according to


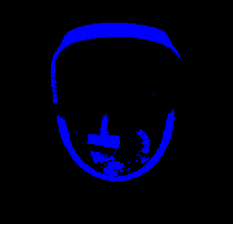




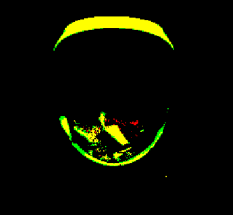
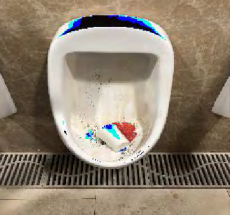

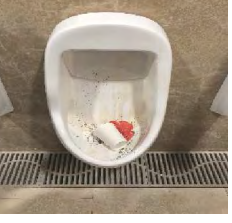



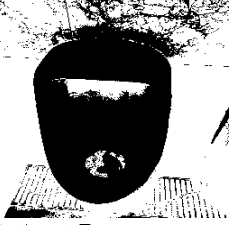


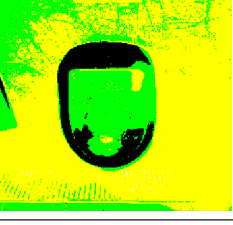

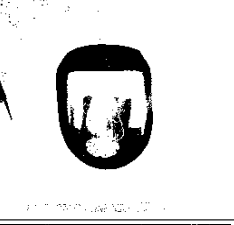


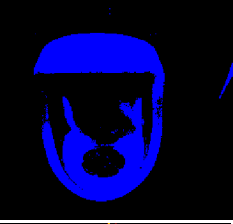

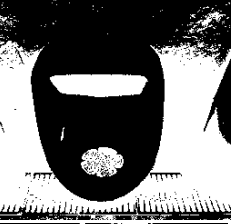


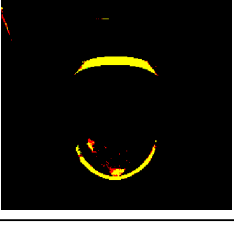

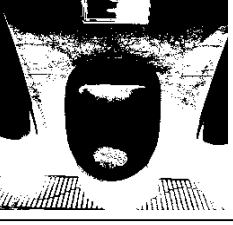

$$g(I_c) = (I_c - I_{cmin}) \frac{max_{new} - min_{new}}{I_{cmax} - I_{cmin}}, \quad (2)$$

$$f_{Normalization}(I) = \begin{cases} g(I_{red}) & c = red \\ g(I_{green}) & c = green \\ g(I_{blue}) & c = blue \end{cases} \quad (3)$$

where I_c denotes the c^{th} color channel (*red, green, blue*) of the given image I (in this scenario $I = S_b^n$), I_{cmin} represents the minimum value of the c^{th} channel, maximum value represents I_{cmax} , max_{new} and min_{new} are the maximum and minimum values of the expected normalized image. In this work, we used $max_{new} = 1$ and $min_{new} = 0$ as normalization boundaries. Normalized image and calculated v matrix was added to obtain the I'_{vec} .

The result I'_{vec} was multiply by a constant β , sampled from a standard normal distribution with a purpose of providing

TABLE 1. Color augmentation results for three categories.

Category	Original image: S_b^n	Color augmentation matrix: $\beta I'_{vec}$	Color augmented image (output): Im_1	Binary thresholding matrix: I_{BW}	Noise suppressed image (output): Im_2
Dirty					
					
Average					
					
Clean					
					

a randomness and to control the intensity of color augmentation. This color augmented matrix $\beta I'_{vec}$ (shown in Table. 1-col 2), is added to the given image S_b^n (shown in Table. 1-col 1) and results depicted as Im_1 (shown in Table. 1-col 3). Thereafter, $\beta I'_{vec}$ normalized back to the 0 to 255 range using eq. 3 and then converted it into a binary

image I_{BW} using

$$f_{BinaryThresholding}(I) = \begin{cases} 0 & I < 128 \\ 1 & I \geq 128 \end{cases} \quad (4)$$

where $I = \beta I'_{vec}$. We used a control variable $\gamma \in \mathbb{R} | 0 < \gamma < 1$ to control the intensity of I_{BW} . The binary thresholding

matrix, I_{BW} is then subtracted from the channels of the given input image S_b^n , the results stored as Im_2 , as shown in Table. 1-col 5. Finally, using only one image (S_b^n) we created two color augmented images Im_1 and Im_2 .

Next subsection focuses on a method of selecting a suitable CNN model for feature extraction.

B. FEATURE EXTRACTION

Over the years, many CNN architectures were introduced for image classification task, and all these models were proven to give an acceptable range of image classification accuracy. Even though these deep architectures perform well in their tested datasets, training such an architecture from scratch requires millions of images and significant computational power. Since our dataset is limited, a suitable pre-trained model needed to be selected for feature extraction. Then arise a question, how to select a pre-trained model suitable for our dataset. In this section, we tackle this problem through a PCA feature analysis process and this assisted us to determine the model that is best suitable for our dataset.

Most of the deep architectures designed and trained to tackle ImageNet's dataset [21]. It contains 1.2 million images in 1,000 categories, but our specific categories are not one of them. However, since these deep architectures have been well studied in the literature and with a proven track record, it will be effective if we can leverage on these deep architectures in solving our problem. The usual process to perform this is to employ several pre-trained models and fine-tune or retrained those using the new training data. Then compare the accuracies between each retrained models and decide the best suitable model for the new dataset. But this process could be highly time consuming and need more computational resources. We have addressed this problem by exploiting PCA on the extracted features from different models without doing any fine-tuning or retraining. We used VGG-16, ResNet-50, and Inception-V3 as the pre-trained DCNN models for our experiment.

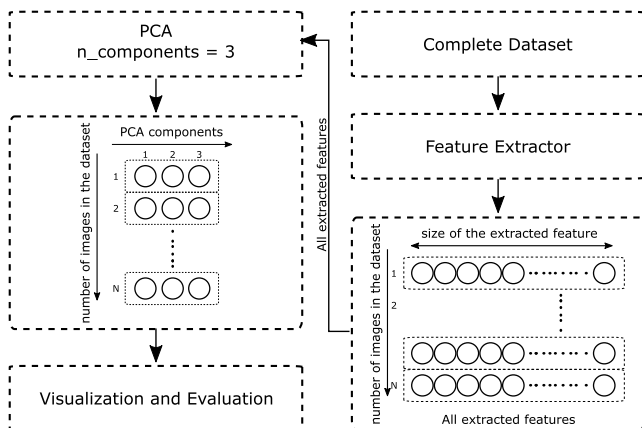


FIGURE 4. The proposed evaluation process to select the suitable architecture for our dataset.

According to the Fig. 4, complete dataset is used in the evaluation process. We removed the soft-max layer and some

fully-connected layers from the pre-trained models and created four different feature extractors. The first feature extractor is created by removing the last two fully connected layers (FC-3 and FC-2) from the VGG-16 and left it only with one fully connected layer (FC-1). Second we used the same VGG-16 and removed just the last fully connected layer (FC-3). Finally in both ResNet-50 and Inception-V3, we dropped layers up to the latter average pooling layer and created the third and fourth feature extractors respectively. Then without performing any retraining or fine-tuning, just use the inference to extract features using the feature extractors mentioned in above. Then perform the PCA on extracted features and reduce the dimensionality up to three components [31]. Thereafter, features are visualized on the calculated principal component space. This process repeated for all four feature extractors separately.

Since PCA can be considered as a clustering method [32], the results from a compatible feature extractor (model) need to demonstrate a cluster formation in PCA space. If PCA results are not showing any form of a cluster or they are completely random, then feature extractor has failed in extracting features over the given dataset. Therefore, no use of utilizing it as a feature extractor as it only introduces a randomness, which can also achieve by a random weight initializer. Hence, more meaningful clusters means model is capable of understanding the dataset. The most suitable model was selected through this process, which is shown in Fig. 4, and then performed fine-tuning using the selected feature extractor. Results of this process discussed in the Section III led us to select Inception-V3 as our feature extractor.

The next section focuses on classifying the data by utilizing the Inception-V3 average pooling layer results as the extracted features.

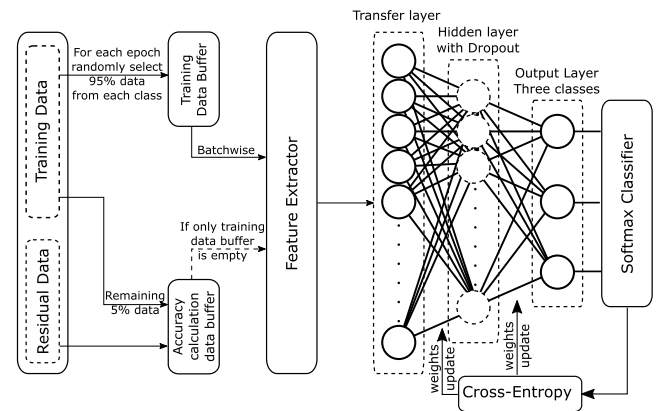


FIGURE 5. Classifier and its training procedure.

C. THE CLASSIFIER AND TRAINING PROCEDURE

We acquired an neural network classifier to classify the extracted features from the pre-trained Inception-V3 in to the given three categories. The numerical values of the flattened average pooling layer of the Inception-V3 feature extractor is taken as the input to the classifier. As depicted in Fig. 5,

the flattened average pooling layer is denoted as *transfer layer* and its numerical values are *transfer values*. However, extracted features or transfer values haven't demonstrated the expected clusters in the PCA space, hence the model requires further target specific feature extraction [33]. Therefore, we used another hidden layer with dropout regularization before the soft-max classifier layer see Fig. 5.

During the training stage, 95% of training data from each category buffered in a *training data buffer*, the remaining 5% and the residual data loaded into a *accuracy calculation data buffer*. This data buffering always occurs before every epoch. In Fig. 5, at the end of every epoch, the accuracy is calculated using the reserved data in accuracy calculation buffer. Completion of an epoch is denoted when the training data buffer becomes empty. The focus of this arrangement is to enforce the model, not to learn noisy features like backgrounds, translations, etc. The training data buffer feeds the data batch wise, then the soft-max with cross-entropy loss is calculated and minimized using Adam-optimizer in the training process. We used polynomial decaying learning rate for our experiment, starting from 0.1 learning rate to 0.00001 learning rate within 100000 epochs. After model started to show compelling results, we terminated the training process and evaluate the trained model using test data Y .

IV. RESULTS AND DISCUSSION

This section discuss about the results of data augmentation, feature extraction, classification, and finally a brief experiment about the model adaptability to other restrooms. Our experiment setups are implemented using TensorFlow framework [34] and python 3.5, on an HP Z840 workstation with NVIDIA GeForce GTX 1080Ti graphics card.

A. DATA AUGMENTATION

Table. 1 show the color augmentation results with respect to the dirty, average, and clean categories. It is evident after observing the color augmentation matrix images, that the proposed algorithm can identify the dirtiness (outliers) exist in urinal bowl. Examining the color augmented images closely, it is apparently visible that the color of the dirty part of images has been enhanced. In the clean category, there has not been too much change because it does not contain any dirtiness (outliers). But if we look closely at the color augmented image in the clean category, some of the sewer line pixels have been augmented, but the overall image remains almost same as the original image. The binary thresholding matrix I_{BW} is able to distinguish the difference between dirty and non-dirty pixels clearly. For instance, the binary thresholding matrices of the average category able to identify the dirtiness very precisely. Noise suppressed images Im_2 were produced by subtracting the binary thresholding matrix I_{BW} by the original image I . Upon comparing the noise-suppressed image and the original image, it is clearly visible that the details of the non-dirty parts are suppressed, even the wall textures and shadows are suppressed to some extent.

Since PCA can act as an outliers detector, we were able to use it to detect dirtiness in images. Therefore, color augmented image Im_1 will enhance the pixels around dirtiness, while Noise suppressed image Im_2 will increase the attention to dirtiness.

B. FEATURE EXTRACTION

The results for each four feature extractors are depicted in Fig. 6. After a thorough observation, we can observe that, except Inception-V3 (Fig. 6d), all others extracted features have either shuffled or randomly distributed throughout the space. However, Inception-V3 extracted features related to the dirty category are almost completely separated from the features of clean and average categories. This indicates that Inception-V3 feature extractor can effectively differentiate dirty and non-dirty features as compared with the other three feature extractors. Therefore, we selected the Inception-V3 up to its average pooling layer as our CNN feature extractor.

The typical method is to build four different end-to-end solutions using the four different feature extractors with the existing classifier or employing a new classifier and compare the results at the end, which probably consume lots of resources and computational time. Whereas, using our method shown in Fig 4, we were able to select the most qualified feature extractor for our dataset at the middle of the process and saved lot of computational time and effort.

C. TRAINING AND CLASSIFICATION

After training the classifier using extracted features from Inception-V3 feature extractor, we performed a PCA cluster analysis on the extracted features of the classifier hidden layer, see Fig. 7b. After comparing Fig. 7a and Fig. 7b, it's clear that the classifier is succeeded in extracting unique features from training data and distinguish between them.

TABLE 2. Comparison of classification performances.

Approach	Precision (%)	Recall (%)	Accuracy (%)	Inference time per image (seconds)
BoF (SURF + kMeans)	55.7	58.3	58.27	40×10^{-3}
HOG + SVM (4x4 Cell)	74.2	74	73.99	0.75×10^{-3}
HOG + SVM (2x2 Cell)	78.6	78.2	78.21	2.2×10^{-3}
HOG + SVM (8x8 Cell)	81.9	81.7	81.72	0.24×10^{-3}
VGG-16	84.63	83.33	84.14	1.885
ResNet-50	85.24	84.71	85.46	3.055
Inception-V3	86.43	85.90	88.10	3.385
Proposed Method	91.3	90	90.52	2.87

Table 2 shows the comparison of classification performances of our proposed method with other state-of-the-art computer vision algorithms. To the best of authors knowledge, there is no any other datasets exist in literature relevant to cleanliness classification. Moreover, our method focus only on classifying cleanliness in a restroom. Therefore, we conduct the performance analysis using our dataset and compared the accuracies. Our proposed method was able to achieve comparatively higher accuracy score than

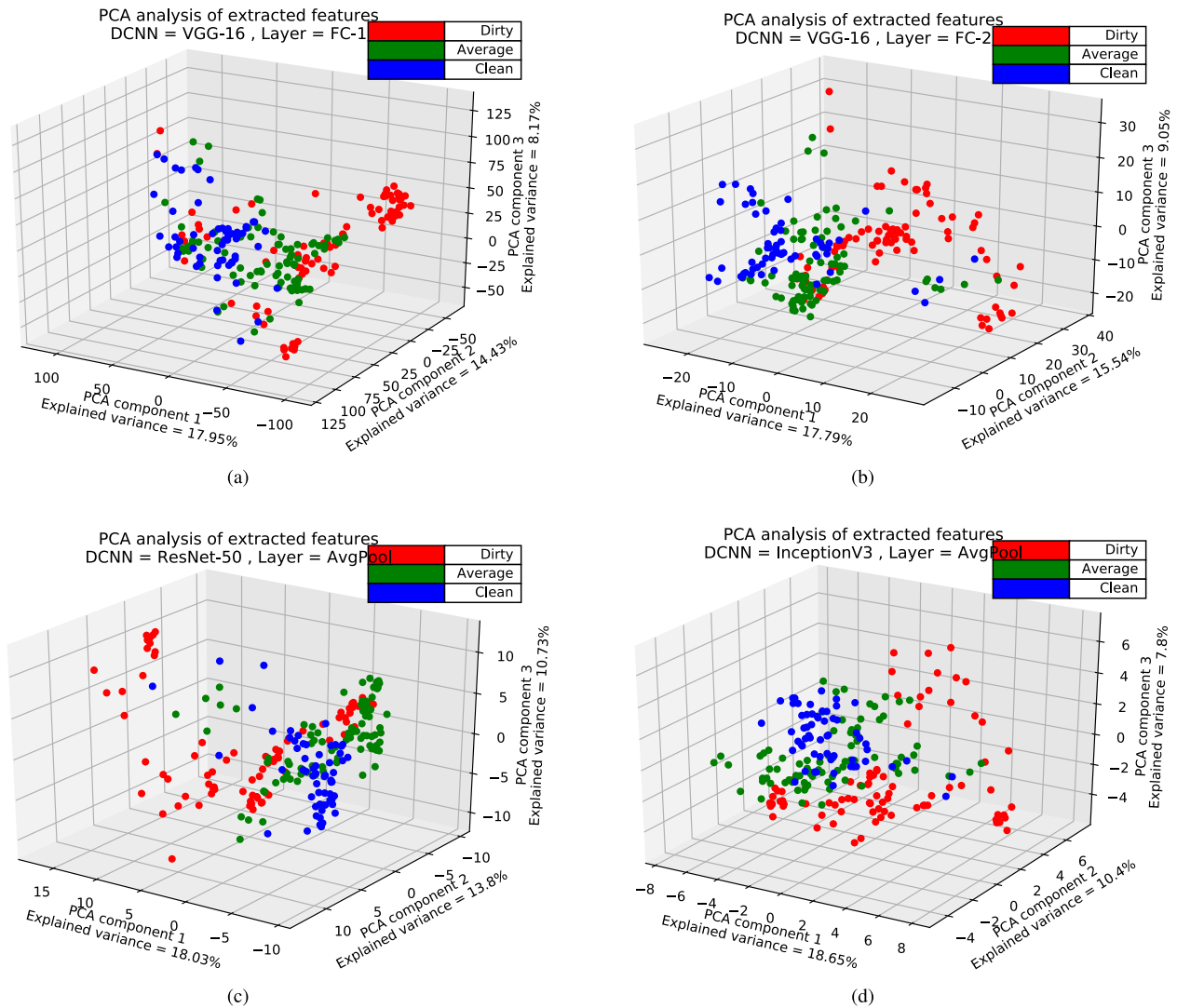


FIGURE 6. PCA cluster analysis for feature extractors are presented. (a), (b) results relate to VGG-16 feature extractors and (c), (d) results relate to ResNet-50 and Inception-V3 feature extractors respectively.

other algorithms. Note that the better performance of the proposed method is limited to the targeted application, as it is specifically design for that. It does not mean that the proposed method can be better than the state-of-the-art classification methods in a general classification problem. However, when considering about inference time, the methods which do not involve convolution operations consume less time than other deep learning methods. Therefore, inference time needs to be taken in to account when deploying the proposed method in a real-time environment.

Table 3 shows the confusion matrix for the proposed method. The confusion matrix (CM) is calculated as,

$$CM(i, j) = 1/|Y_{class_i}| \sum_{y \in Y_{class_i}} P(class(\hat{y}) = class_j | y) \quad (5)$$

where $CM(i, j)$ denotes the prediction of j^{th} class when given an i^{th} class element, y represent an element of i^{th} class and $class(\hat{y})$ denotes the predicted class label. For some data,

TABLE 3. Confusion matrix for the proposed method.

Category	Dirty	Average	Clean
Dirty	0.99	0.00	0.08
Average	0.02	0.83	0.15
Clean	0.00	0.04	0.95

the classifier yields accurate class with a low probability. Therefore, even for an accurate result, the classifier decision uncertainty could be high. Evaluating the performances using Eq. 5 will reflect those uncertainties in the results. According to Table 3, it's clear that the average category has the lowest prediction accuracy. For our proposed method, this result can be validated using Fig. 7b. Some features of the average category seem to merge with the clean category features as

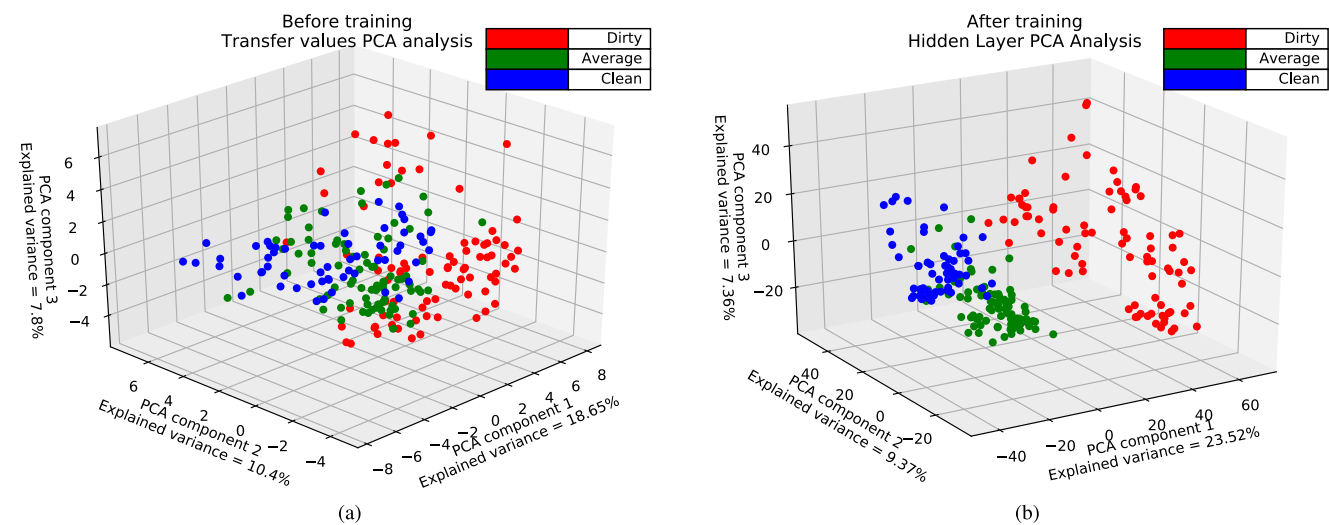


FIGURE 7. Comparison of PCA cluster analysis between extracted features of the classifier before and after the training.

TABLE 4. Some selected results to show the trained model adaptability for other restrooms.

Dirty	0.923	0.981	0.435	0.172	0.482	0.00
Average	0.002	0.003	0.005	0.009	0.517	0.001
Clean	0.074	0.014	0.559	0.733	0.007	0.997
Result	Dirty (Accurate)	Dirty (Accurate)	Clean (Inaccurate)	Clean (Accurate)	Average (Accurate)	Clean (Accurate)

well as the dirty category boundary. Because of that, the classifier showed a low predicted probability for the average category.

Next, we gathered images from some other restroom facilities and some online urinal bowl images as our new test dataset *Y*. Then we utilized this new test dataset to check our trained model adaptability to a new restroom environment. Table 4 shows some results reported for various types of restroom conditions and urinal bowl shapes. After observing the images closely, the dirty images were identified except the third image. But the inaccurate result doesn't have a higher prediction value. Even though these images contain various types of backgrounds, shapes, lighting conditions, elevations, etc., our proposed methodology was able to classify them accurately up to an expectable level. This is a promising result to make sure that our model has the potential to adapt into a new environment. However, further studies are required to ensure such robustness.

V. CONCLUSION

In summary, this paper has developed a machine learning inspired solution for cleanliness classification in restroom facilities by learning important features using the proposed methodology. We have devised an color augmentation algorithm to enhance and introduce dirtiness to our urinal bowl images, a method to identify a qualified neural network architecture and its transfer layer for feature extraction, and a competent classifier along with its training procedure. These all were application specific implementations to achieve our goal of classifying cleanliness into dirty, average, and clean categories. Finally, we had a promising results of our architecture's ability to perform in different other restrooms environments. The limitations of this method could be highlighted as the difficulty of collecting data, sometimes cameras could have blind spots and dirtiness could depends on lighting conditions and other environmental facts. However, despite these limitations, our approach demonstrated

acceptable results. Therefore, as our future work, these limitations will be investigated and further experiments will be carried out to evaluate the model adaptability in various types of other restrooms. Furthermore, in depth study will be conducted towards explaining the model decisions using Explainable AI (XAI) techniques [35].

REFERENCES

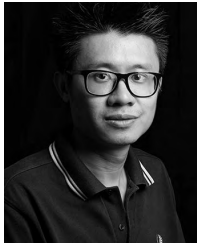
- [1] Singapore National Environment Agency. (2012). Cleaning service industry Cleaning performance for Commercial Premises SS499. Accessed: Dec. 16, 2018. [Online]. Available: <https://www.nea.gov.sg/our-services/public-cleanliness/cleaning-industry/cleaning-industry>
- [2] Smart Technologies. (2016). Restroom Visitizer System. Accessed: Dec. 16, 2018. [Online]. Available: <http://smarttechnologies.sg/>
- [3] NCS. (2016). Smart Toilet Analytics. Accessed: Dec. 16, 2018. [Online]. Available: <http://www.ncs.com.sg/documents/20184/73669/NCS+Smart+Toilet+Analytics.pdf>
- [4] SmartClean Technologies. (2017). Virtual Cleaning Supervisor. Accessed: Dec. 16, 2018. [Online]. Available: <http://www.smartclean.sg/>
- [5] W. Liu, M. Zhang, Z. Luo, and Y. Cai, "An ensemble deep learning method for vehicle type classification on visual traffic surveillance sensors," *IEEE Access*, vol. 5, pp. 24417–24425, 2017.
- [6] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
- [7] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proc. Int. Workshop Multimedia Inf. Retr.*, 2007, pp. 197–206.
- [8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
- [9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. CVPR*, Jun. 2005, pp. 886–893.
- [10] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.
- [11] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proc. NIPS*, 2007, pp. 801–808.
- [12] S. Gao, I. W.-H. Tsang, and L.-T. Chia, "Kernel sparse representation for image classification and face recognition," in *Proc. ECCV*, 2010, pp. 1–14.
- [13] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. CVPR*, Jun. 2006, pp. 2169–2178.
- [14] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2559–2566.
- [15] B. A. Draper, K. Baek, M. S. Bartlett, and J. R. Beveridge, "Recognizing faces with PCA and ICA," *Comput. Vis. Image Understand.*, vol. 91, nos. 1–2, pp. 115–137, Jul./Aug. 2003.
- [16] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. CVPR*, Washington, DC, USA, Jun./Jul. 2004, p. 2.
- [17] L. Jayasinghe, T. Samarasinghe, C. Yuen, J. C. N. Low, and S. SamGe, "Temporal convolutional memory networks for remaining useful life estimation of industrial machinery," [Online]. Available: <https://arxiv.org/abs/1810.05644>
- [18] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375–9389, 2017.
- [19] R. Xi et al., "Deep dilation on multimodality time series for human activity recognition," *IEEE Access*, vol. 6, pp. 53381–53396, 2018.
- [20] M. Alhussein and G. Muhammad, "Voice pathology detection using deep learning on mobile healthcare framework," *IEEE Access*, vol. 6, pp. 41034–41041, 2018.
- [21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, Jun. 2009, pp. 248–255.
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.
- [23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition." [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Jun. 2016, pp. 770–778.
- [25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. CVPR*, Jun. 2016, pp. 2818–2826.
- [26] Y. Lin et al., "Large-scale image classification: Fast feature extraction and SVM training," in *Proc. CVPR*, Jun. 2011, pp. 1689–1696.
- [27] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *Proc. ECCV*, 2010, pp. 143–156.
- [28] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proc. ECCV*, 2014, pp. 346–361.
- [29] M. Lin, Q. Chen, and S. Yan. (2013). "Network in network." [Online]. Available: <https://arxiv.org/abs/1312.4400>
- [30] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
- [31] S. H. Marakkalage et al., "Understanding the lifestyle of older population: Mobile crowdsensing approach," *IEEE Trans. Computat. Social Syst.*, to be published, doi: [10.1109/TCSS.2018.2883691](https://doi.org/10.1109/TCSS.2018.2883691).
- [32] S. Raychaudhuri, J. M. Stuart, and R. B. Altman, "Principal components analysis to summarize microarray experiments: Application to sporulation time series," in *Proc. Symp. Biocomput.*, 2000, pp. 455–466.
- [33] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. NIPS*, 2014, pp. 3320–3328.
- [34] M. Abadi et al. (2016). "TensorFlow: Large-scale machine learning on heterogeneous distributed systems." [Online]. Available: <https://arxiv.org/abs/1603.04467>
- [35] A. Binder, G. Montavon, S. Lapuschkin, K.-R. Müller, and W. Samek, "Layer-wise relevance propagation for neural networks with local renormalization layers," in *Proc. ICANN*, 2016, pp. 63–71.



LAHIRU JAYASINGHE received the B.Sc. degree (Hons) from the Department of Electronic and Telecommunication Engineering, University of Moratuwa, Moratuwa, Sri Lanka, in 2016. He then worked in the industry for one year in the field of electronic design automation. Then, he joined the Singapore University of Technology and Design, Singapore, as a Researcher. His current research interests include predictive maintenance analysis, deep learning, computer vision, and pattern analysis methods for practical problems.



NIPUN WIJERATHNE received the B.Sc. degree from the Department of Electronic and Telecommunication Engineering, University of Moratuwa, Moratuwa, Sri Lanka, in 2016. He then joined the Singapore University of Technology and Design, Singapore, as a Researcher. His current research interests include machine learning, deep learning, signal processing, scientific data mining, and pattern analysis methods for practical problems.



CHAU YUEN (S'02–M'08–SM'12) received the B.Eng. and Ph.D. degrees from Nanyang Technological University, Singapore, in 2000 and 2004, respectively. He was a Post-Doctoral Fellow with Lucent Technologies Bell Labs, Murray Hill, NY, USA, in 2005. From 2006 to 2010, he was with the Institute for Infocomm Research, Singapore, as a Senior Research Engineer, where he was involved in an industrial project on developing an 802.11n wireless LAN system and participated actively in

long term evolution (LTE) and LTE Advanced standardization. He joined the Singapore University of Technology and Design, Singapore, in 2010. He holds two U.S. patents and has authored or co-authored over 300 research papers in international journals or conferences. He was a recipient of the IEEE Asia-Pacific Outstanding Young Researcher Award, in 2012. He serves as an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS and the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.



MENG ZHANG was born in Shandong, China, in 1964. He received the bachelor's degree in industrial electrical automation engineering from the China University of Mining and Technology, in 1986, the master's degree in bioelectronics engineering from Southeast University, Nanjing, China, in 1993, and the Ph.D. degree in microelectronics. Since 1993, he has been a Faculty Member with the National ASIC System Engineering Technology Research Center, Southeast

University, where he is currently a Professor. He has authored or co-authored more than 30 papers cited in SCI and EI, and the holder of more than 40 Chinese patents and two U.S. patents. He has been a member of the program committees for IEEE conference.

• • •