Interpretation of Deep CNN Recognition with Filter Space Clustering in Feature Extraction and Reconstruction

Sukhan Lee
Sungkyunkwan University,
Suwon 440-746, South Korea
lsh1@skku.edu

Naeem Ul Islam
Sungkyunkwan University,
Suwon 440-746, South Korea
naeem@skku.edu

Abstract

Interpreting a deep Convolutional Neural Network (CNN) involves identifying the features in a hierarchy of layers that contribute to recognition. Although the current approaches serve as methods to interpret a deep CNN, further advancement is required for a more accurate and efficient way of understanding how a hierarchy of features formed by a deep CNN contributes to recognition. In this paper, we propose attaching a feedback CNN to a pretrained feedforward CNN as a means of learning how recognition is performed by the feedforward CNN. In other words, the features reconstructed in a hierarchy of the feedback CNN represent those learned by the feedforward CNN. By analyzing how clusters are formed in the layers of feature spaces in the feedback CNN, we can interpret which features critically contribute to recognition. It also helps to evaluate whether or not recognition is done successfully. In order to show this, we experimentally verify the capabilities of the proposed approach in terms of identifying incorrectly recognized input data by pinpointing the source of the error in feature spaces. Experiments conducted on the ModelNet datasets indicate that the proposed approach offers an extended capability of interpreting a deep CNN as described above with higher accuracy than conventional approaches.

1. Introduction

Deep neural networks offer tremendous benefits under the available resources, however, this end-to-end training process with highly nonlinear functions of deep networks treats them as black boxes which lack proper information about the internal representation of the data. The activities of neurons toward the representation of the internal structure of the data as well as their behavior in terms of collaboration with each other in such complex models are obscure, and the model learning is based on trial-and-error. This is a substantial limitation of deep networks in understanding the classification applications, as it hinders the human experts in carefully verifying the classification decision. In summary, the unreasonable properties of assessing the model based on the binary or real-valued one-dimensional answer at the decision layer make it hard to interpret the activities of neurons at different layers. Furthermore, the black-box nature of deep networks makes it nearly impossible for one to know about the collaboration of neurons or fix problems when errors occur while performing different tasks. Therefore, meaningful interpretation of a deep neural network is required which allows the user to learn the behavior of the network and trust its ability in terms of interacting with the system using deep networks in different applications.

In this paper, we take advantage of the clustering-based evaluation of the recognition probabilities by reconstructing them from the selected cluster to the testing samples while using feature extraction and reconstruction CNN.

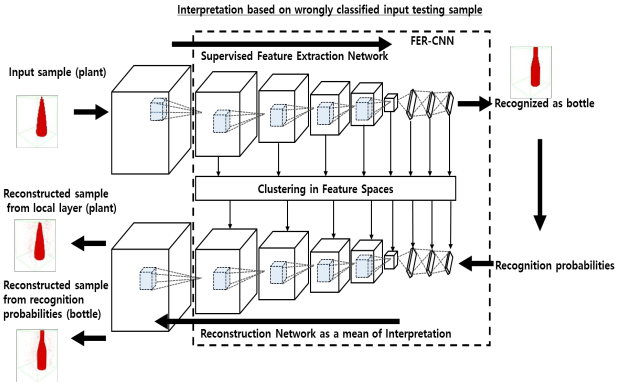

Figure 1: FER-CNN model, the top represents the encoder part, whereas the bottom represents the decoder part along with skip connections between the encoder and decoder. Both the encoder and decoder have five convolution layers and deconvolution layers, respectively, along with three fully connected layers.

2. Feature extraction and reconstruction CNN

For automatic local and global feature extraction along with automatic reconstruction, we use feature extraction and reconstruction (FER-CNN) as a basic platform as shown in Fig. 1 for the purpose of interpretation.

FER-CNN is composed of two sub-networks: the Encoder and Decoder. Both the Encoder and Decoder are mirrored, where each consist of five convolution layers with the parameters of 6x6x6, 5x5x5, 4x4x4, 3x3x3, 2x2x2, and 1x1x1, as well as fully connected layers with dimensions of

1500, 500, and 10/40. The filter dimensions are [64, 196, 512, 1024, 2048, 1500, 500] respectively.

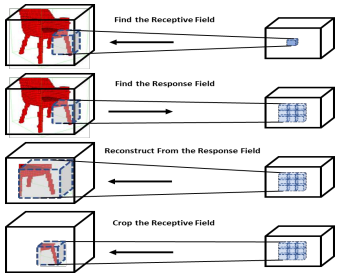

Figure 2. The first row shows the feature location in the feature space which is mapped back to the input space, the second row shows the response field from the corresponding input receptive field, the third row shows the reconstruction from the effective response field, and the last row represents the required receptive field.

3. Response field-based reconstruction

In order to assess the performance of the deep convolutional neural network, the activities of each neuron in the intermediate layers must first be known. FER-CNN is composed of convolutional layers, so reconstruction from a single feature is not straightforward. The receptive field in input space is not only the function of the given feature in that particular layer, but is also dependent on the neighboring features in that spatial location. This is due to the fact that overlapping convolution windows are dictated by the choice of layer configuration (i.e. window size and stride). In order to address this problem, we use a response field-based reconstruction algorithm, which is presented in Fig. 2.

Algorithm 1 Response Field-Based Reconstruction Algorithm

1. Select a location at a specific layer for which we need to find the response field.
2. Find the receptive field for that specific response in the given layer.
3. Calculate the response field in that layer for all of the neurons in the input space.
4. Copy that response field and make the rest of the feature space zero.
5. Reconstruct the input space from the computed response field at that particular layer.
6. Crop the receptive field obtained in the second step.

4. Implications of feedback weights as a mean of interpretation

The proposed feedback network with its weights trained based on the input dataset plays an important role in interpreting what is learned in the feedforward network to which it is attached. The reconstruction of an input sample through the feedback network connotes the way in which the feedforward network clusters a hierarchy of features for classification. As such, the reconstruction ignores the noise deformation of input samples but generates more typical representatives corresponding to the classification result. FER-CNN provides an effective means for achieving the desired goal of interpreting the cluster formed in the feedforward path of the classification network, where the feedback layers preserve information about the individual samples locally as well as their mean representation globally. The clustering-based interpretation explores the effect of feedback weights on the interpretation of deep neural networks, where we first cluster the feature space in each layer along the filter dimensions using the k-means clustering algorithm. As we used the ModelNet10 dataset, which has 10 classes, for this analysis, we set k=10 in the last fully connected layer. For the rest of the layers, we set k= [60,50,40,30,30,20,20] as appropriate. Upon completion of the clustering of the feature space, we select the testing samples and pass them through the network, then find the nearest cluster in each layer at each location, as shown in Fig. 3. The representative clusters at each layer are implicit in the feedforward path, as shown in the odd rows of Fig. 3. By contrast, the even rows represent the representative clusters in the feedback path, which are more biased to the typical representation of the input sample, and contains information about their representative classes which we explain further in the following section.

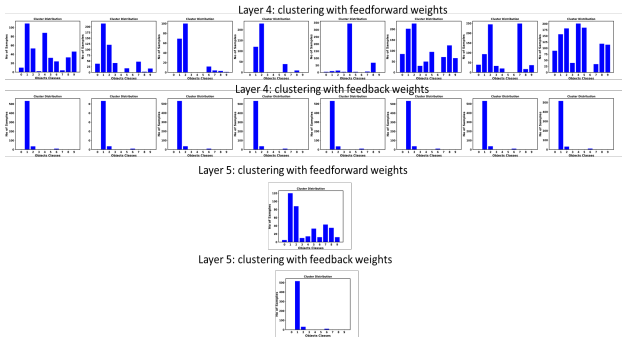

Figure 3. The implication of feedback weights as a mean of the interpretation of deep neural network using clustering-based analysis. The top row represents layer-wise representative clusters in the feedforward path while the bottom row represents the layer-wise representative clusters in the feedback path.

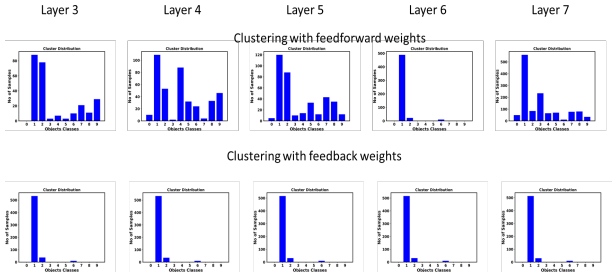

Figure 4. Clustering-based analysis of the correctly classified input testing samples from layers 3 to 7. The first column

represents the clustering in the feedforward direction. The second column represents the clustering in the feedback direction.

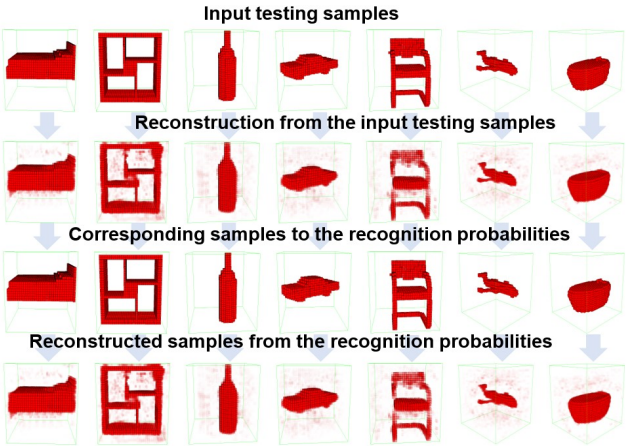

Figure 5. Clustering-based analysis of the correctly classified input testing samples by reconstructing them from the local layers as well as from the correctly classified class probabilities. The first row shows the input testing samples and the corresponding columns in the second row represent the generated samples from the local layers, while the third and fourth rows show the nearest ground truth and their corresponding reconstructed samples to the cluster centroid based on the recognition probabilities, respectively

4.1. Clustering based interpretation and FER-CNN reconstruction

In order to perform the clustering-based analysis for the interpretation of a deep neural network, we cluster the feature space of all of the layers along the filter dimensions using feedforward and feedback weights. First, we cluster the feature space of all of the layers along the filter dimensions using the feedforward parameters of the network, then pass the testing sample through the network. In this step, we select the cluster centroid at each spatial location nearest to the corresponding location in the input testing sample and then pick the selected clusters as representatives of that testing sample. The cluster distribution at the local layers provides us with information about the features which are more biased to the intraclass feature similarities than the specific object class. The cluster distribution becomes more representative of that class as we go up the network, and the last layer cluster represents the class of that object. Second, we cluster the feature space of all of the layers along the filter dimensions using the feedback parameters of the network. After clustering the feature space in the feedback path, we pass the input testing sample through the network, then obtain its code at each layer using the feedback parameters. Then, using the KNN algorithm, the nearest representative cluster centroids at each layer and each spatial location to the corresponding spatial location in the input testing sample are obtained.

The clusters formed in the feedback path carry information about the misclassified samples to the input space as a mean of qualitative and quantitative analysis for the interpretation of the network. Figs. 4 to 7 shows the interpretation of the deep neural network based on the above discussion. First, we analyze the correctly classified input testing sample using the cluster-based interpretation shown in Fig. 4. The first row shows clustering in the feedforward direction whereas the second row shows the clustering in the feedback direction. Each column in Fig. 4 represents a layer-wise analysis. This analysis indicates that the selected clusters are specific to the corresponding class based on the above discussion. In the second analysis, we used the wrongly classified input testing sample, and the candidate clusters obtained from the feedback weights in the designated layers represent the wrong class of the objects as shown in the second row, whereas the feedforward candidate clusters show the correct class corresponding to the input testing sample, as shown in the first row of Fig. 6. The qualitative analysis for the interpretation of the deep neural network using feedback weights is shown in Figs 5 and 7. In Fig. 5, we first analyze the samples which have been correctly classified by the recognition network. As the first step of the analysis, we provide these input testing samples to the network then reconstruct these from the local layers using the feedback weights. As discussed previously, the local layers are more biased to the feature similarities than the classification objective; therefore, it reconstructs the samples having similarities with the input testing samples. The reconstructed input testing samples are shown in the corresponding columns in the second row in Fig. 5. The recognition network generates the class probabilities based on the input testing samples. These class probabilities are fed to the feedback network, which generates their features at each layer. The features at each layer then pick the nearest cluster centroids and reconstruct them. The nearest sample to the cluster centroid and its corresponding reconstructed results are shown in the third and fourth rows of Fig. 5, respectively, and the results indicate that the feature similarity-based reconstructions from the local layers as well as from the generated class probability match each other, hence, proving the correct classification accuracy. For example, the bed, bookshelf, bottle, car, chair, airplane, and bathtub are all correctly classified by the recognition network and their reconstructed results from the correct class codes show similarities with the input testing samples, as shown in Fig. 5. In Fig. 7, we analyze the samples which have been wrongly classified by the recognition network. During this analysis, the network reconstructs these input testing samples from the local layers using the feedback weights as shown in the corresponding columns in the second row of Fig. 7, which shows similarities to the input testing samples regardless of their classification. Based on these input testing samples, the

recognition network generates the class probabilities. These class probabilities are fed to the feedback network, which generates their features at each layer. The features at each layer then pick the nearest cluster centroids and reconstruct them. The nearest sample to the cluster centroid and its corresponding reconstructed results are shown in the third and fourth rows of Fig. 7, respectively. The results show that the samples reconstructed from the local layers match the input testing samples regardless of their recognition probability, whereas the samples generated from the cluster centroids using the class probabilities as input to the feedback network do not match the input testing samples, and it selects the clusters belonging to the recognition probabilities generated by the network. This analysis further elaborates that the samples which are wrongly classified show structural similarities to the input testing samples. For example, the bathtub, cone, table, bookshelf, bottle, car, and chair are wrongly classified as a bed, bottle, bathtub, bench, cone, bookshelf, and bed, respectively. The misclassified samples show structural similarities with the input testing sample. Such analysis provides insight into the classification of the recognition network by qualitatively interpreting its recognition.

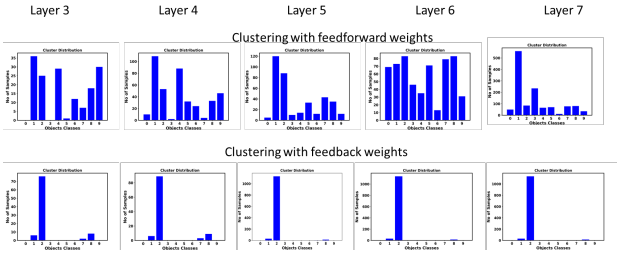

Figure 6. Clustering-based analysis of the wrongly classified input testing samples from layers 3 to 7. The first column represents the clustering in the feedforward direction. The second column represents the clustering in the feedback direction.

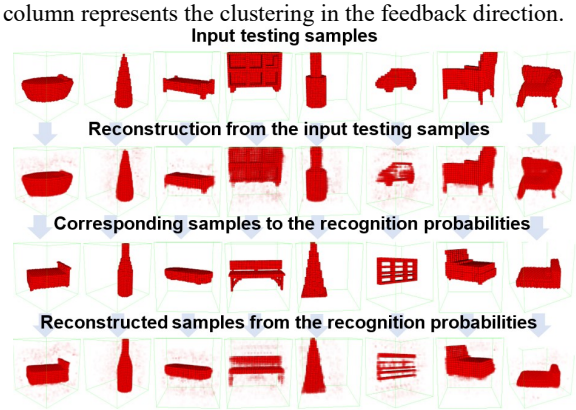

Figure 7. Clustering-based analysis of the wrongly classified input testing samples by reconstructing them from the local layers as well as from the correctly classified class probabilities. The first row shows the input testing samples and the corresponding columns in the second row represent the generated samples from the local layers, while the third and fourth row show the nearest ground truth and their corresponding reconstructed samples to the cluster centroid based on the recognition probabilities, respectively.

We also performed quantitative analysis in terms of the mean square error of the misclassified input testing sample and the generated sample from the wrong recognition probabilities presented in Table. I. This analysis shows that the error between the input testing sample and the reconstructed sample from the misclassified class probability is almost double the error between the input testing sample and its reconstruction from the local layers. This error difference provides information about the misclassification of the input testing samples.

TABLE I
MEAN SQUARE ERROR BETWEEN THE INPUT SAMPLE AND RECONSTRUCTED SAMPLES.

Error between input representative and reconstructed output	19.185
Error between the misclassified sample and reconstructed output	22.610172
Error between reconstructed input representative sample and reconstructed misclassified	26.466608
Error between the input representative and reconstructed misclassified sample	35.837

5. Acknowledgement

This work was supported in part by the "3D Recognition Project" of the Korea Evaluation Institute of Industrial Technology (KEIT) under Grant 10060160, and in part by the "Project of e-Drive Train Platform Development for small and medium Commercial Electric Vehicles based on IoT Technology" of Korea Institute of Energy Technology Evaluation and Planning (KETEP) (20172010000420).