EE 569: HOMEWORK #6:
Feedforward CNN Design and Its Application to the MNIST Dataset

Problem 1

(a) Understanding of feedforward-designed convolutional neural networks

1.a.1 Abstract and Motivation

Convolutional Neural Network (CNN) involves the processing of an input image of higher dimension with feature extraction and classification simultaneously to an output vector of lower dimension equal to class label. In case of Backpropagation based CNN, the model parameters are trained/optimized continuously until the cross entropy of the output classification is minimized. To decrease the training complexity and to increase the interpretability of CNN, Feedforward CNN is adopted. It involves the model/network training in a single epoch using data-centric approach.

1.a.2 Approach and Procedure

The architecture of Feedforward CNN (FF-CNN) is similar to the construction of layers in LeNet-5, except the operations involved at each layer.

- Feedforward based CNN is a single/one pass unlike Backpropagation which changes the filter coefficients based on gradient descent optimization for minimum cross entropy.
- Each of the input, output or intermediate transformations are represented as a vector space.
- Each of the vector representation involves transformation from high dimension to low dimension (sub-sampling layer and fully connected layer) or low dimension to high dimension (convolution layer). The training data ensures to find the transformation and the testing data follows this transformation.
- Transformations make use of
  a. Dimension reduction through subspace approximation with adjusted/added bias for convolutional and sub sampling layer.
  b. Sample clustering and classification using pseudo labels for fully connected layers.
- Convolutional layer: As the spatial domain reduces, the loss of information is compensated by adding/ increasing the spectral component. The pixels in the spatial filter are projected onto anchor vectors obtained by principal component analysis.
- The process involves dimensionality reduction. But reducing the dimension leads to loss of information. Hence, the reduced dimension should have large discriminant power to differentiate different feature/classes. Convolution involves increasing the

dimension (more filters are incorporated) followed by subsampling (max pooling) decreasing the dimension. These operations accelerate to choose the dimension which have large discriminant power.

- Fully connected layer: Subdivided into feature class, sub-class space and class-space. The features obtained from consecutive convolutional layer and sub sampling layer is not directly reduced to output layer (10 class for MNIST). But, are first reduced from 120D to 84D and 84D to 10D to increase the discriminability power and to obtain diverse in inter-class variability, pseudo labels are created.
- Computational neuron:
  $y_k = a_k^T x + b_k$ or $y_k = \sum_{n=0}^{N-1} a_{kn} x_n + b_k$ where k is the number of filters (spectral), N is the size(length) of the local receptive filter and $z_k = \emptyset(y_k) = \max(0, y_k)$
  a_k is the filter coefficients while training the network and is referred as anchor vector while testing. The filter coefficients can be viewed as a match filter/correlator and y_k is the response obtained when a patch of the input (x) of a given window size is coupled with the filter a_k. z_k is a non-linear activation function which bypasses only positive responses ($y_k \geq 0$) and clips the negative responses ($y_k < 0$) to zero.
- In case of BP-CNN, the filter coefficients a_k is learned/updated continuously through backpropagation and chain rule for minimizing the cross-entropy loss at the output node for multiple epochs until the convergence of gradient descent. Thus, involving higher training complexity.
- In case of FF-CNN, Saab (Subspace approximation with adjusted bias) transform is incorporated for the choice of filter coefficients a_k over a single epoch.
- No information can be gained by looking at a single pixel, hence a local receptive filter is used for convolution to identify the feature from a larger view point.
- Since FF-CNN is over a single epoch (one-pass manner), the features must be identified correctly with stronger discriminant power to distinguish between different classes.
- For the choice of filter coefficients (a_k) and the bias term (b_k) Saab transform is used.
- A subspace in R^N is spanned by the anchor vectors a_k as follows:
  a. $a_0 = \frac{1}{\sqrt{N}} (1, 1, 1, \ldots, 1)^T$ spans DC subspace
  b. $a_1, a_2, \ldots a_{k-1}$ anchor vectors span the AC subspace which is complement to DC subspace. N is equal to window size of the receptive filter (ex: if the receptive filter size is 5x5, then N is 25D), K is the number of spectral filters and is usually less than N. A covariance matrix of the input vector x (a high dimensional vector of length N) is computed and the eigen vectors corresponding to largest (k-1) eigen values are the vectors which span the AC subspace. Equivalently, principal component analysis on AC subspace determines the (k-1) anchor vectors with strongest discriminant power (the first k-1 orthogonal component). Principal component analysis can be interpreted as pooling in spectral domain.
- Covariance matrix for an input image of size NxN is N^4, which involves lot of complexity. Therefore, input corresponding to smaller window size N is considered for covariance matrix computation.
- Bias selection (b_k):

a. Positive Response constraint

The selection of b_k must yield a positive response (y_k)

$y_k = \sum_{n=0}^{N-1} a_{kn} x_n \geq 0$

b. Constant Bias constraint

For all 'k' spectral filters, the choice of b_k is constant. Since, it acts as a DC component, it can be removed easily in further layers and the interpretation is simpler. It lies in the DC subspace even if multiple layers are cascaded and it does not affect the selection of anchor vectors which are in the AC subspace. The analysis would be different and difficult in case the bias was chosen to vary for each of the spectral filter (it would be considered as AC component then).

$b_0 = b_1 = b_2 = ... = b_{k-1} = d\sqrt{K}$

The single bias term 'd' for all the spectral filter is chosen using the bias selection rule:

Bias Selection Rule:

$b_k \geq \max \|x\|$ k=0, 1, … , k-1 so that $d \geq \frac{1}{\sqrt{K}} \max \|x\|$ (since the inner product / correlation can be -1 at max)

- With both positive response and constant bias constraint, y_k (response) is guaranteed to be positive, eliminating the necessity of the non-linear activation function (ReLU- Rectified Linear Unit). Since, the output remains the same with or without the activation function), this accounts to a simpler design.
- Saab transform is better than Saak transform since there is no necessity of introducing more spectral filters and the sign confusion problem could be eliminated by adjusting the bias term.

- Fully connected Layer: The functionality of the fully connected layer is to transform the spatial- spectral output from the convolutional, sampling layer to an output vector predicting the class label

  In LeNet-5 architecture, 3 fully connected layer are involved to transform 5x5x16 D = 400 D to 120 D, 120 D to 84 D and 84D to 10 D. But for the input to the fully connected layer only 5x5x15=375D is considered, since the first spectral filter accounts for the mean, which is removed.
- FF- CNN uses Linear least square regressor for each fully connected layer unlike BP-CNN which trains the coefficients using backpropagation and chain rule for gradient descent optimization of minimum cross-entropy of the output vector (class label).
- The reason there exist 3 fully connected layer instead of 1 is:

  a. Performance is better

  b. Inter-class variability (diversity) is included.
- For training/ choosing the coefficients of the fully connected layer, least square regression is used as follows:

$$
\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & w_1 \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & w_2 \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} & w_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{c1} & a_{c2} & a_{c3} & \cdots & a_{cn} & w_c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \\ 1 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_c \end{bmatrix}
$$

Where x is the input from each node of FC layer and y is the expected output from each output node of FC layer (usually one-hot encoder/vector) and w_1 ,..w_c are bias terms of the linear regressor.

- Dimensionality reduction from n to c using FC layer, may not have any expected output (y: one-hot encoder/vector) always since it might be a hidden layer. A pseudo label is created by k-means clustering(k=Q) (ex: 0 label is further divided into 0-i, 0-ii, 0-iii … 0-xii etc for dimension reduction from 5x5x15 to 120D). Therefore, combining the class label and the auxiliary cluster label, a pseudo label is created and the expected output for each node of the fully connected layer is derived and fed as an input to find the coefficients.
- A pseudo cluster label is created to increase the interclass variability or the diversity at each layer. Clusters for each of the class label are created. Within each class, there are multiple sub-label created.
- Cascade of multiple Linear least square regressor or Multistage linear least square regressor is to align the feature space into output vector (predicting the class label) with least cross-entropy.

Similarities and differences between FF-CNN and BP-CNN

- To avoid backpropagation and to obtain robust feature extraction: FF-CNN
- One pass
- Input statistics to design
- Transparent and interpretable
- Data diversity
- Ensemble model

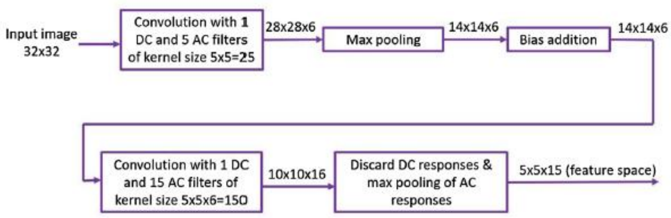


Fig. 5. Summary of the FF design of the first two convolutional layers of the LeNet-5.

Problem 2

(b) Image reconstructions from Saab coefficients

1.b.1 Abstract and Motivation

Image reconstruction is an important aspect of image processing. The convolutional neural network fails to reconstruct the image back since it uses the non-linear activation function. While the 'SAAK' transform was successful in reconstructing the image back. But 'Saab' transform is preferred since 'saak' transform uses double the spectral filter compared to 'saab' transform and the performance is high as well.

1.b.2 Approach and Procedure

SAAK transform:

- **SAAK transform makes use of the ReCOS inverse transform to reconstruct an image back from the convolutional layer.**
- **The number of spectral filters are doubled in case of the SAAK compared to the number of spectral filters of the convolutional neural network (classical)**
- **By doubling the number of filters, both the positive/negative coefficient and its inverse (negated) filter coefficients were as well considered**
- **With the help of sign to position format, the obtained filter co-efficient could be easily identified since the non-linear activation function nullifies the effect of the negatively correlated features**
- **The only cost from the 'SAAK' transform was the disadvantage of using double the filter banks and not using the non-linear activation function**

SAAB transform:

- **It uses the same number of filter size as that of the convolutional neural network**
- **Subspace approximation and adjusted bias transform is a data centric approach and it minimizes the approximation loss by finding a subspace in the input subspace so that the energy is as well preserved.**
- **Principal component analysis (PCA) is made use to find the direction of maximum variance so that the error can be minimized as much as possible.**
- **PCA is used to find the filter co-efficients so that it is best approximated to the input space by dividing the filter bank into DC and AC sub components**

- **The DC component is subtracted from the input samples and PCA is applied to the AC sub component by removing the feature mean and the expectation**
- **The bias is chosen such that a non-linear activation function is absolutely not necessary and the bias is adjusted such that the output is positive**
- **The bias is kept as constant so that it is not varied while reconstructing so that its effect can be removed later.**
- **The performance is better to reconstruct an image since the approximation loss as well as the rectification loss is minimized.**

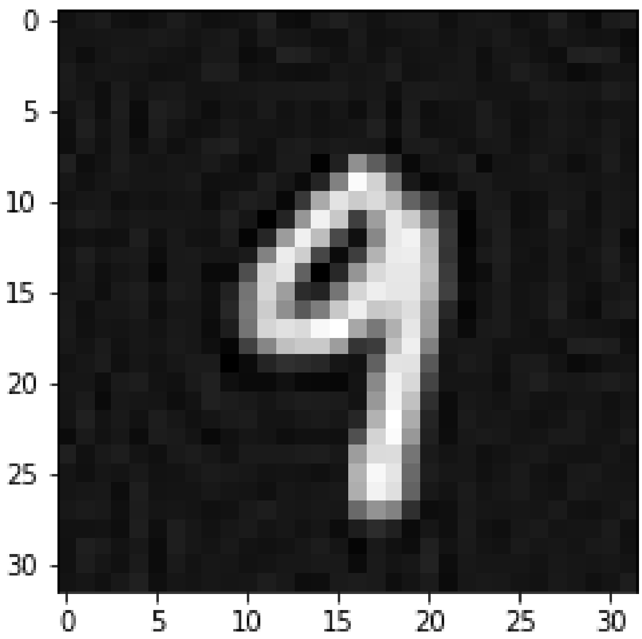
Algorithm:

Step 1: From the obtained feature vector obtained as ‘saab’ coefficient
Step 2: ‘Bias’ of the second stage is added back
Step 3: The single dimensionality of the feature vector is reshaped according to the 4x 16x K1 as there are 4 patches each of dimension 16xK1 where K1 is the spectral filter of first stage.
Step 4: Multiply it with the pseudo inverse transpose of the kernel co-efficients of the second stage
Step 5: Add mean obtained while performing PCA to each sample of second stage
Step 6: Add feature expectation to each of the columns obtained while performing PCA to this stage
Step 7: Add the DC component which was first subtracted before performing AC analysis by PCA
Step 8: Reshape the obtained matrix as 8x8XK1 where K1 is the filter obtained after Step 7
Step 9: Collect the sample patches of size 4x4xK1 and stack them together
Step 10: Multiply it with the pseudo inverse transpose of the kernel co-efficients of the second stage
Step 5: Add mean obtained while performing PCA to each sample of first layer
Step 6: Add feature expectation to each of the columns obtained while performing PCA to this stage of first layer
Step 7: Add the DC component which was first subtracted before performing AC analysis by PCA of first layer
Step 8: Reshape the obtained matrix as 32x32 which is the input image.
Step 9: The maximum pixel intensity is 255. Calculate the mean square error (MSE) between the reconstructed and the original image and obtain the PSNR

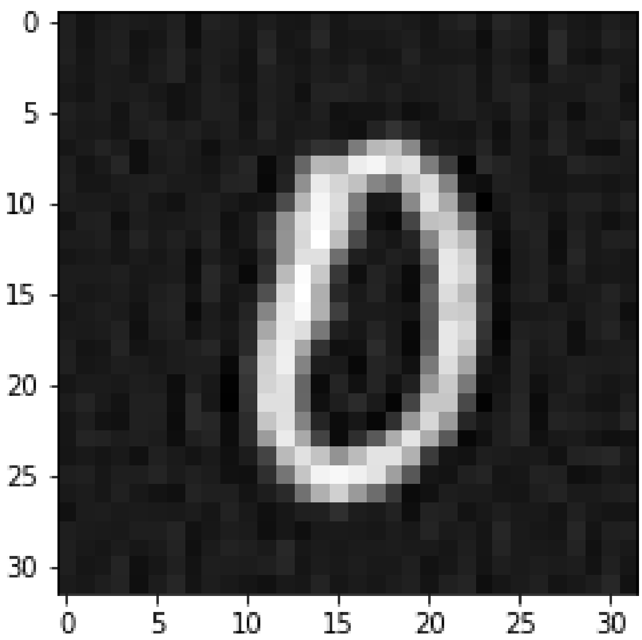
PSNR= 10*log(Max.^2/MSE)

1.b.3 Results

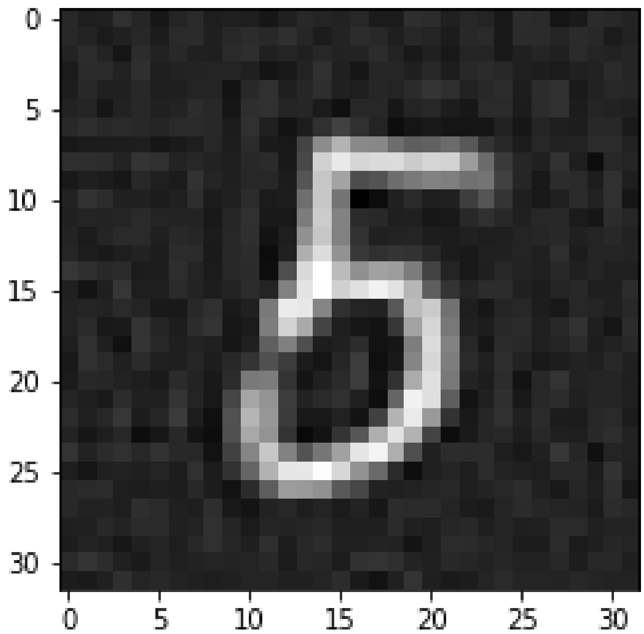
Case 1: 11,81



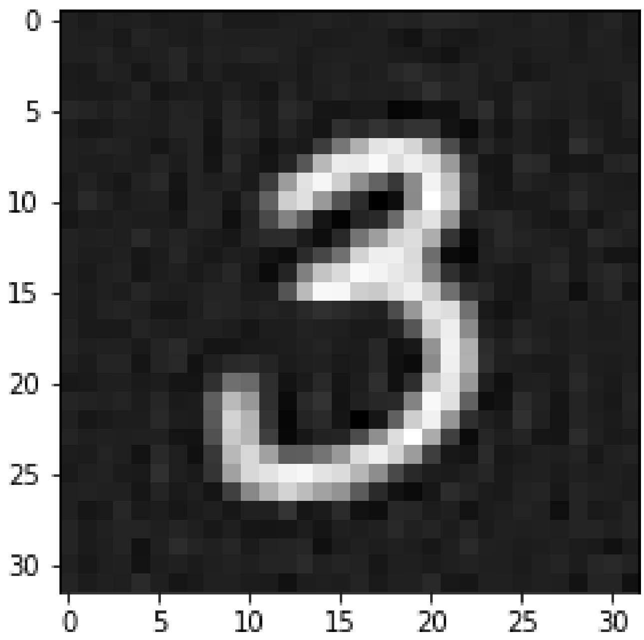
27.786527418297766



25.39926871206363



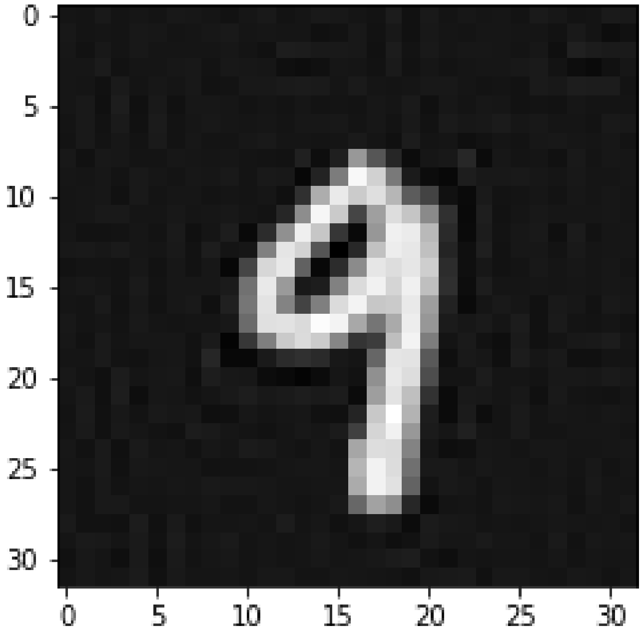
24.03943621983551



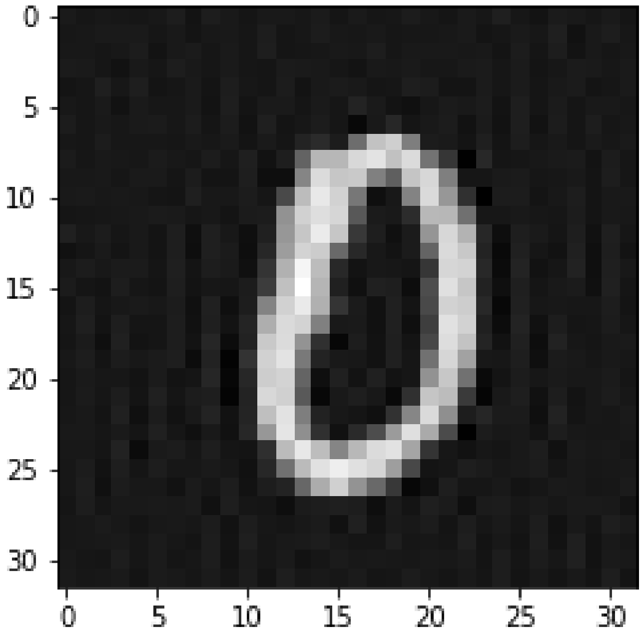
26.25081986080909

Case2: 11,121

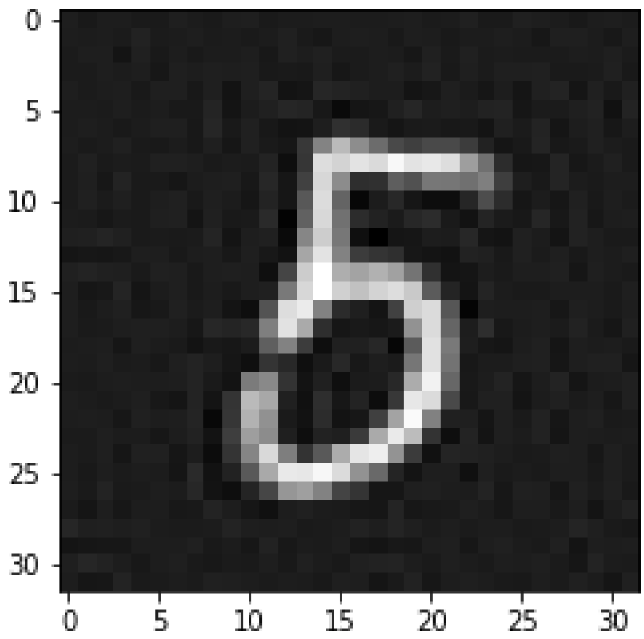
Energy percent: 0.987274



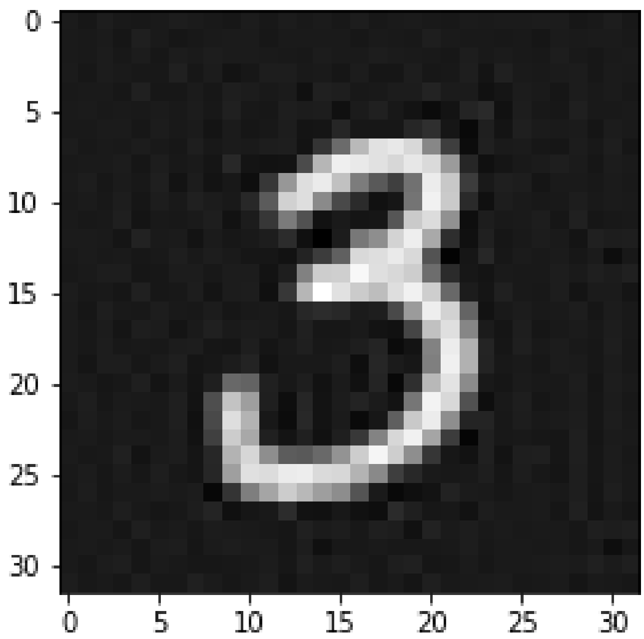
30.292258635342307



27.66383383154888

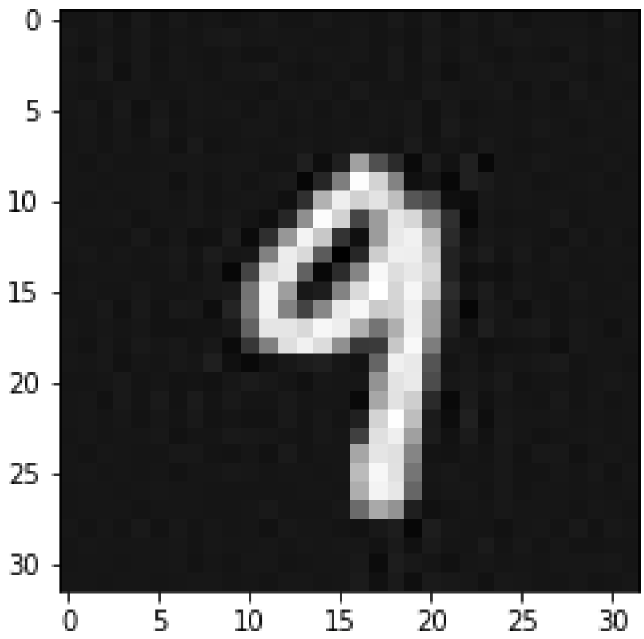


27.566104096454392

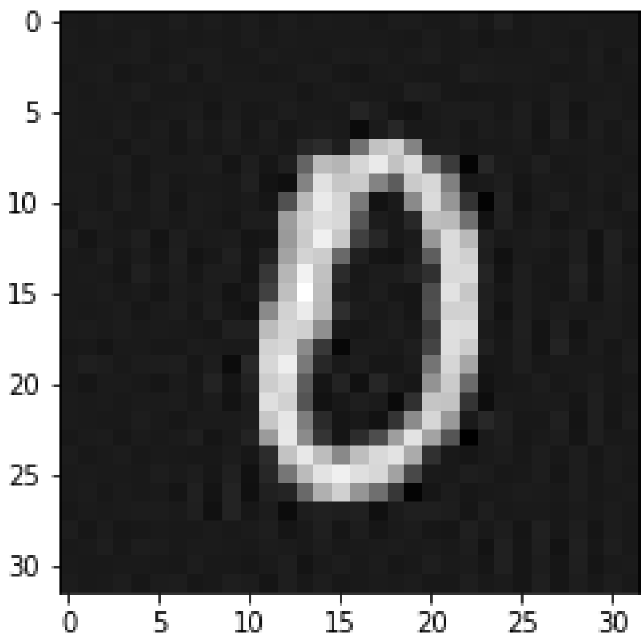


28.644320214773575

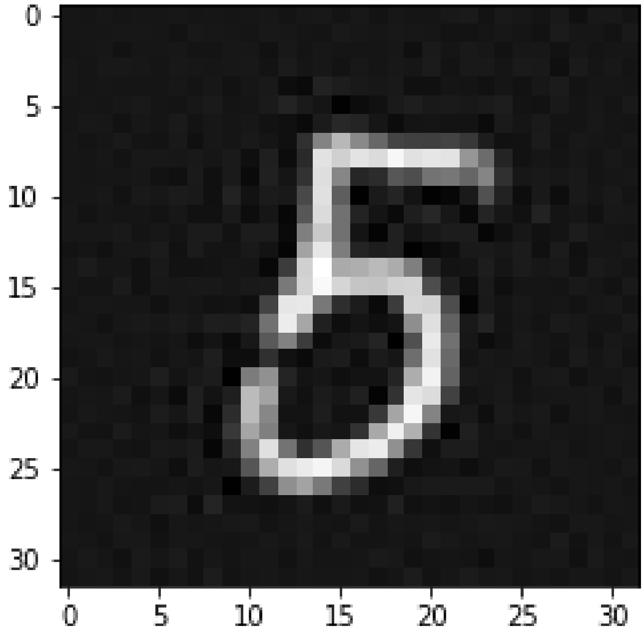
Case 3: 11,141



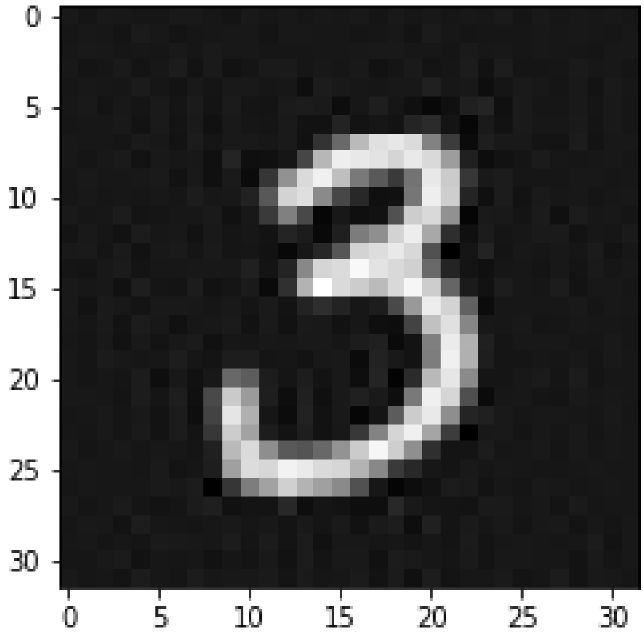
31.888517769734428



28.841313712815655

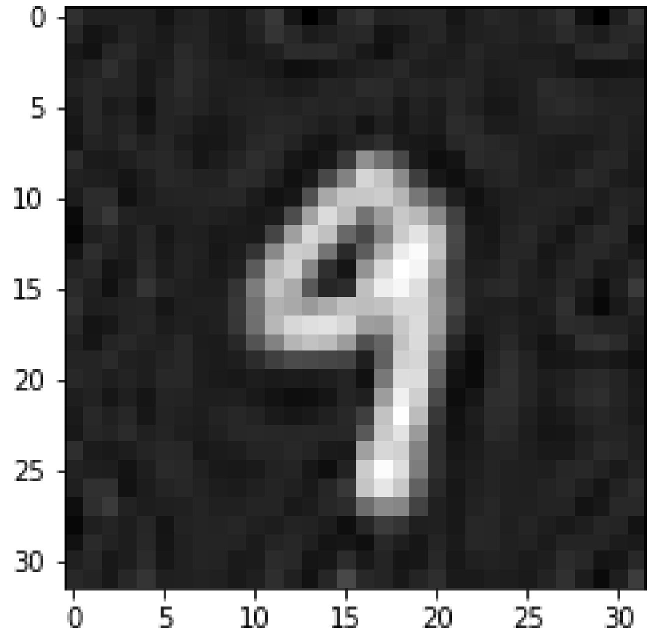


28.74367389982381

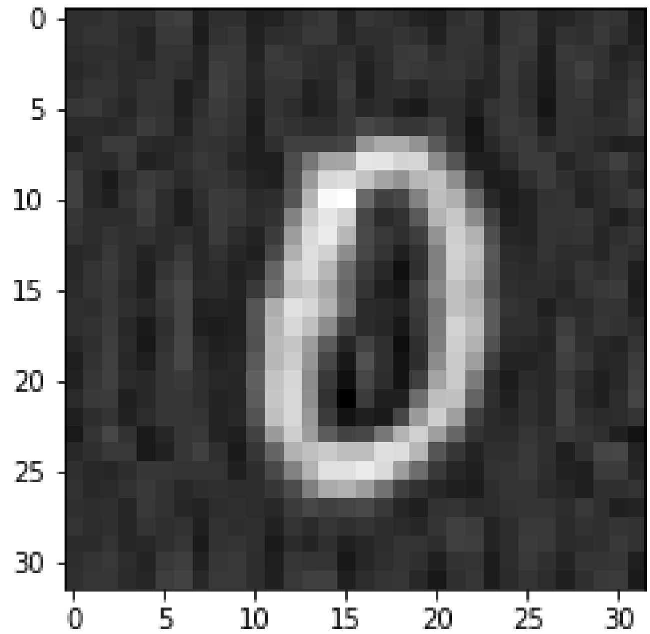


29.1899038253475

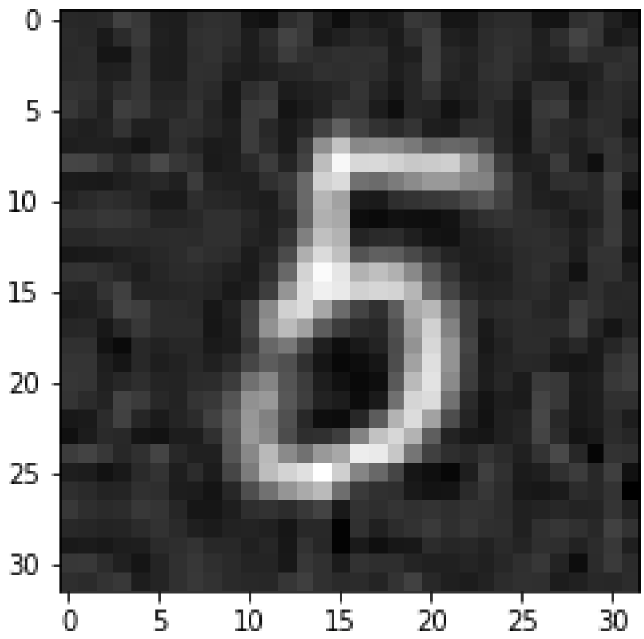
Case 4: 12, 40



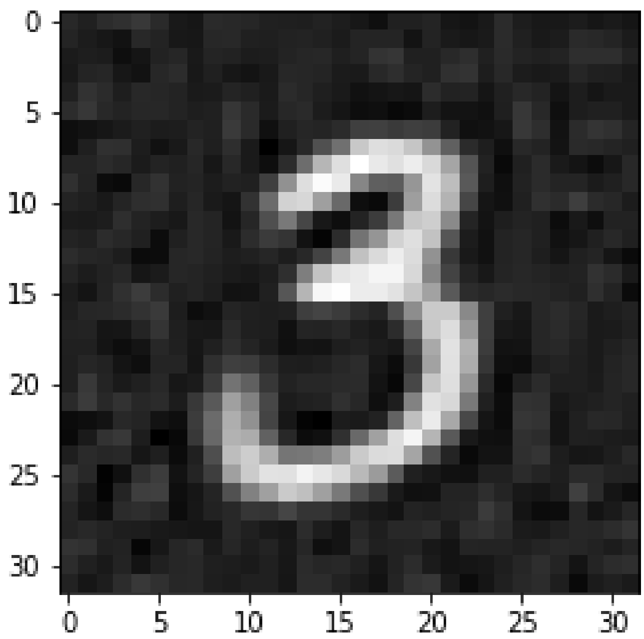
23.35059843239833



21.171362779385973

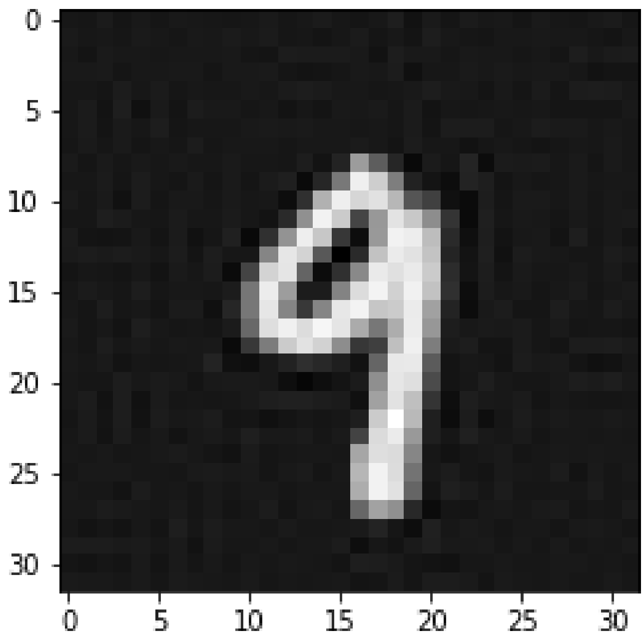


20.836024011607734

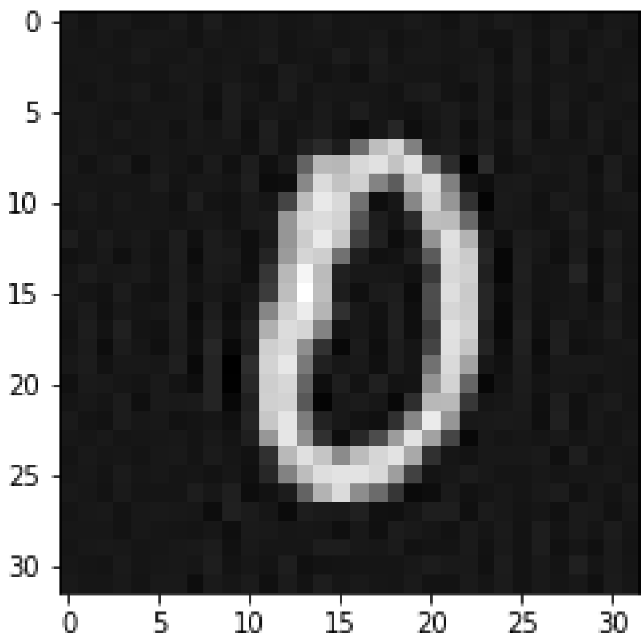


22.366939211269695

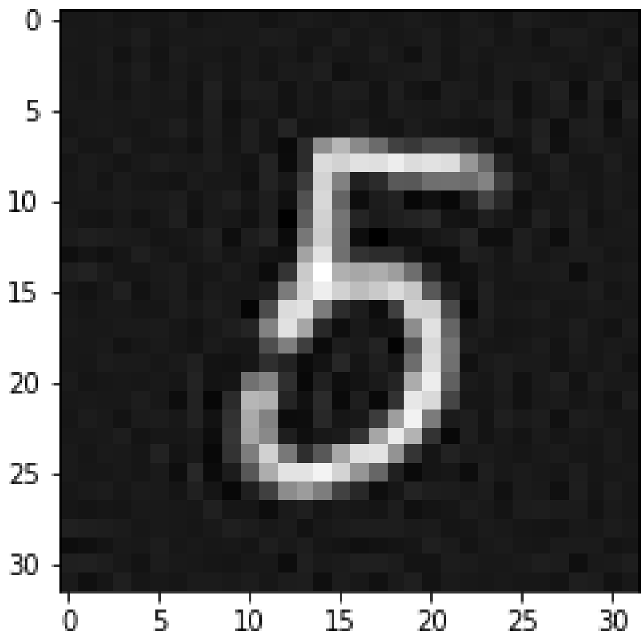
Case 4: 12,130



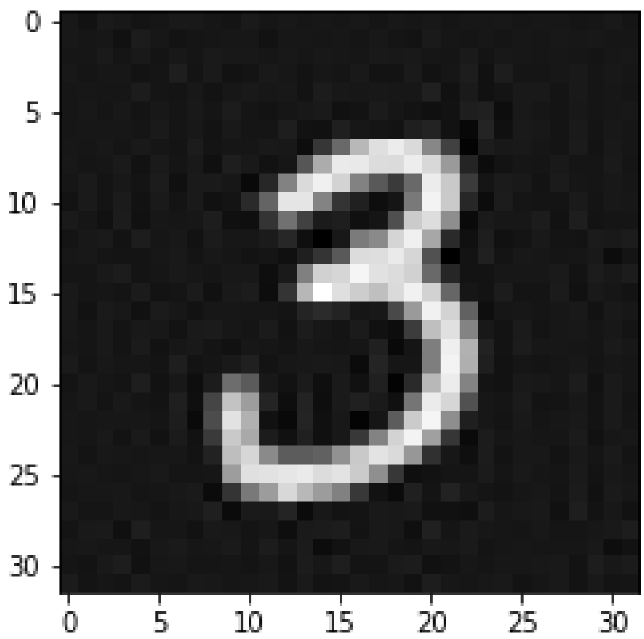
30.939857324625386



28.260729821911585

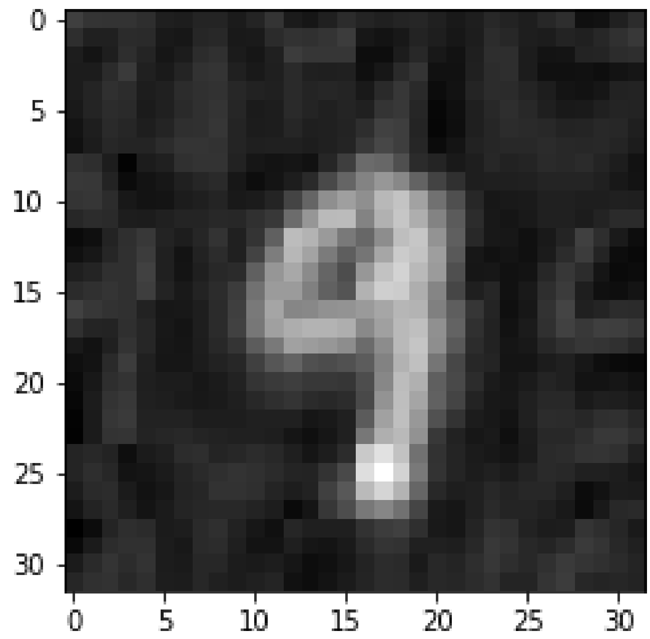


28.412807697795515

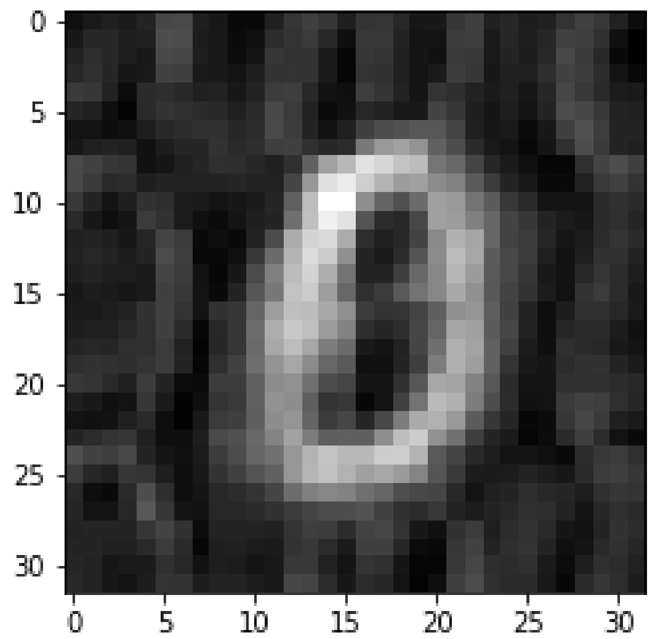


29.651538666972932

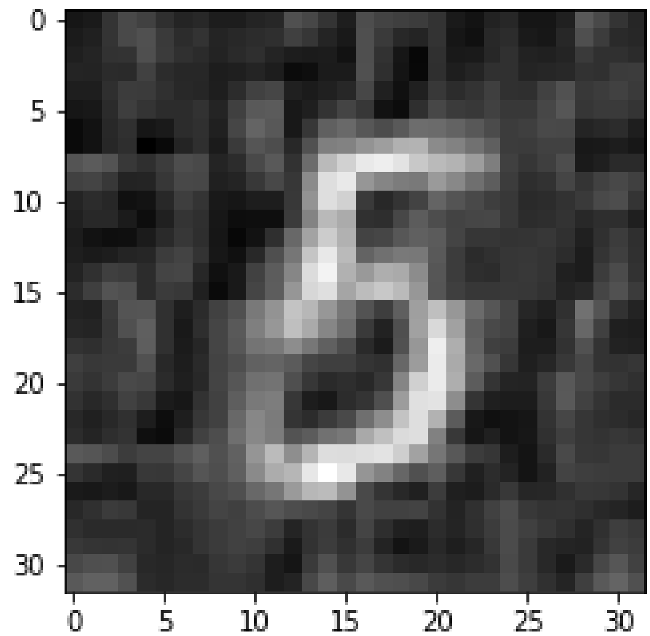
Case 5: 6,16



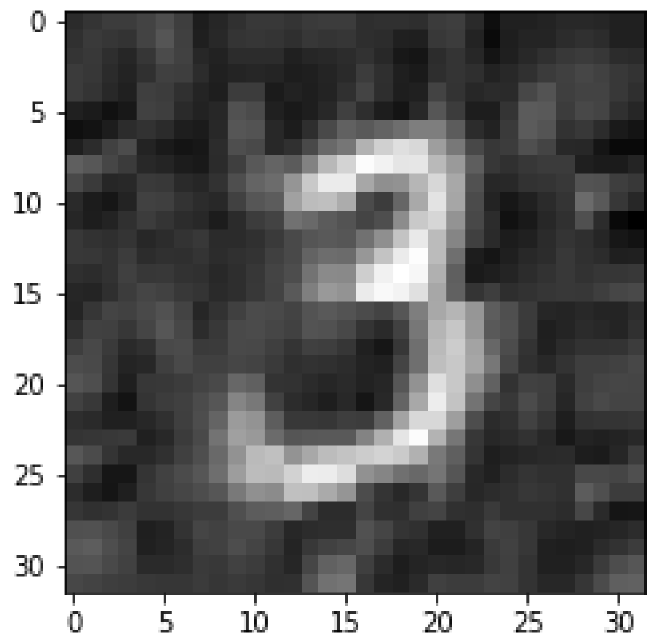
19.01645801868617



16.820003572558548

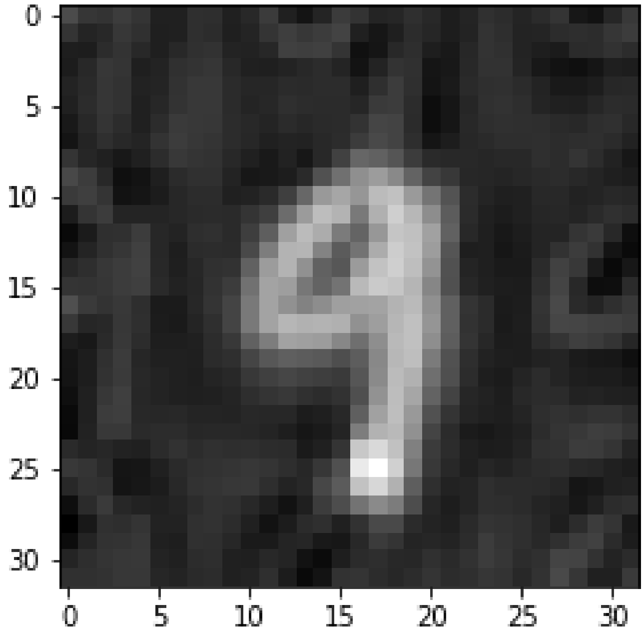


17.709590515762606

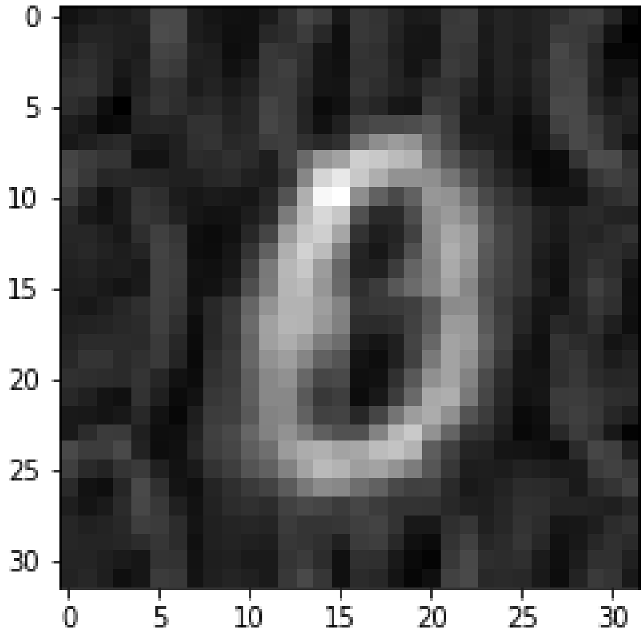


17.679921739555

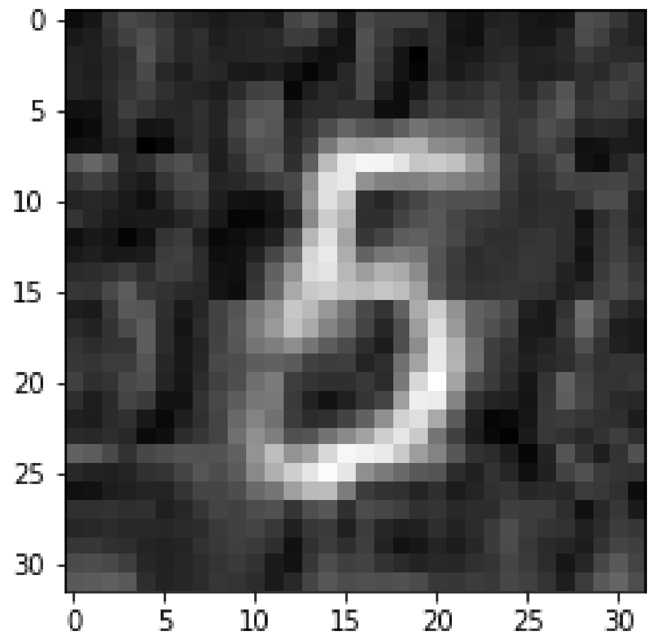
Case 6: 8,16



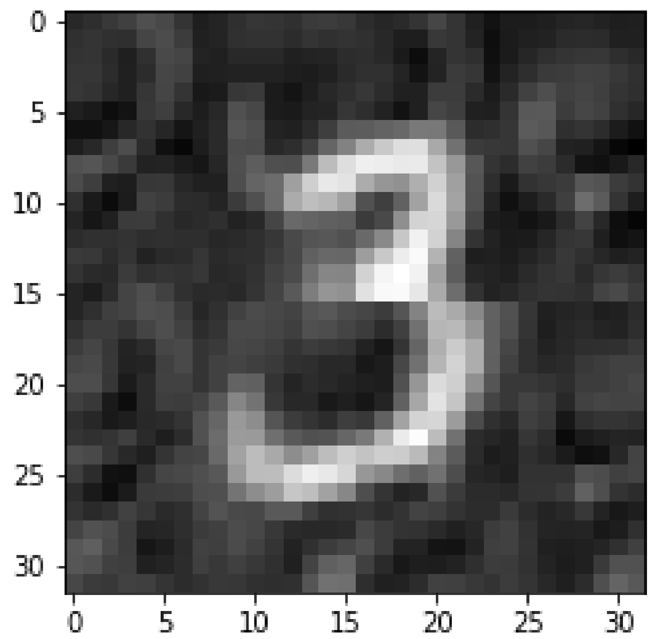
19.22589749163893



16.909579285188546



17.874395869406595



17.836231417201105

1.b.4 Discussion

Cases	**9**	**0**	**5**	**3**
Case 1 11,81	30.292258635342307	25.39926871206363	24.03943621983551	26.25081986080909

Case 2 11,121	30.292258635342307	27.66383383154888	27.566104096454392	28.644320214773575
Case 3 11, 141	31.888517769734428	28.841313712815655	28.74367389982381	29.1899038253475
Case 4 12,130	30.939857324625386	28.260729821911585	28.412807697795515	29.651538666972932
Case 5 8,16	19.22589749163893	16.909579285188546	17.874395869406595	17.836231417201105

From different cases it can be inferred that:

1. Spatial size is 4x4 which is fixed, therefore the dimensionality of first stage filter is 16D and the dimensionality of second stage filter is 16xK2 where K2 is the number of spectral filter in stage 2.
2. Therefore since AC layer is taken into consideration by reducing the dimensionality from 16 and 16*K2 respectively, the weights of the filter co-efficients are to be chosen such that the approximation loss is minimized
3. The eigen vector corresponding to the largest eigen values are chosen from the covariance matrix of the sample patches which are chosen as the filter coefficients.
4.From the table it can be inferred that as the filter number is close to 16 and 16*K2, the PSNR is high.
5. The reason for the same is because while performing PCA, the energy is conserved or preserved.
6. While performing PCA, the eigen vector are chosen in the direction of maximum variance so that the inter-class variability is minimized and the system performs better in classification.
7. Case5 has the best performance and the case 4 has the least performance against the chosen subcases.

Problem 3

(c) Handwritten digits recognition using ensembles of feedforward design

1.c.1 Abstract and Motivation

Feedforward convolutional neural network (FF-CNN) performs slightly low compared to Back propagation convolutional neural network. The reason being that the FF-CCN has a weak classifier. To enhance the performance, an ensemble model is used by increasing the diversity of the input by which it is trained and better performing Support Vector Machine classifier.

1.c.2 Approach and Procedure

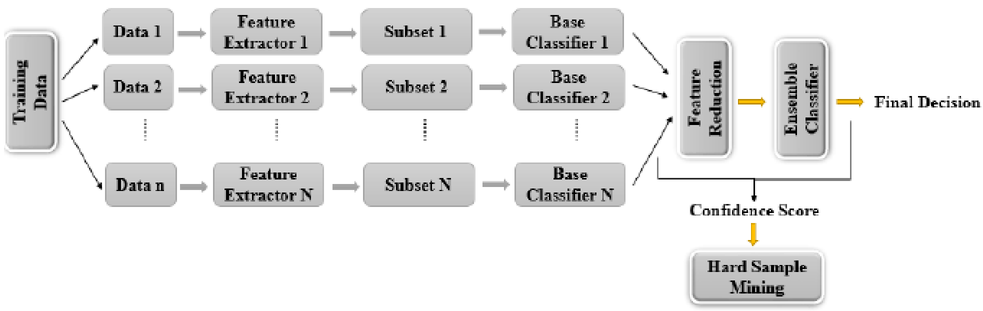


Fig. 1. Overview of the proposed FF-CNN ensemble method.

Block Diagram representation of the FF-CNN ensemble model

- Representation of the data increases the diversity of the input to a greater extent.
- The model is trained under various constraints so that the neural network is capable of generalizing the idea to classify the input image
- The proposed model is trained with more data set with different categories of ‘easy’, ’hard’ obtained by the decision confidence scores.
- By learning with more data and different representations, the system outperforms with respect to all other neural network.
- Base classifiers are the FF-CNN used with different architecture settings
- Diversity is added into the system by three strategies:
  a. Different parameter settings in the convolutional layer: By changing the spatial and spectral domain filter sizes
  b. Flexible feature subsets fed as an input to the fully connected layer: to randomly select the features obtained from the convolutional layer stage1 and 2 to feed it as an input to the ensemble model system
  c. Multiple image embeddings of the same input sources: Different image forms like RGB, Lab color space are used. Images applied to Laws filter can as well be fed as an input to the ensemble model

- The identification of the input samples as ‘easy’, ‘hard’ categories will help in identifying the maximum inter-class variance and training them by providing more robust architecture to have a minimized error rate. The categories define the amount of ease with which the data can be handled/classified
- Ensemble classifiers combines the weak classifier to make them a stronger classifier
- There may exist feature dimensionality reduction in spatial domain which is essential. Channel wise PCA is applied to reduce the spatial domain dimensionality

Algorithm:

Step 1: Collect the feature coefficients in each of the layer from pca_parameters in from Get_Kernel
Step 2: Collect all the features corresponding to the output of the convolutional layer by using Get_features
Step3: Collect all the features, weights,bias corresponding to the fully connected layer by using Get_weights
Step 4: Training accuracy is obtained
Step 5: Step 1 to 4 is repeated for different (10) setting by changing the spatial filter size, spectral filter size and data representation
Step 6: Ensemble all the features from the getweight and perform PCA to reduce the dimensionality of 60K x 100 features to 60K x 60 dimension and SVM classifier is used to test both the training and the testing accuracies. (60K is the training data set)

1.c.3 Results

Spatial Kernel size (Stage 1)	Spectral filter (Stage 1)	Spatial Kernel size (Stage 2)	Spectral Filter (Stage 2)	Training accuracy	Testing accuracy
5x5	6	16	5x5	0.9863	0.97
3x3	6	16	5x5	0.9948	0.9813
5x5	6	16	3x3	0.9925	0.96966
3x3	6	16	3x3	0.9956	0.9743
3x3	10	24	5x5	0.999	0.9903
3x3 Level*Level filter	6	16	5x5	0.99566	0.99233
3x3 Level*Ripple filter	6	16	5x5	0.99263	0.9763
3x3 Spot*Edge	6	16	5x5	0.992	0.988
3x3 Spot*Wave	6	16	5x5	0.991	0.9866
3x3 Wave*Ripple	6	16	5x5	0.9933	0.99633

TRAINING AND TESTING ACCURACIES USING ENSEMBLE MODEL FOR 10 different settings and by using linear SVM classifier

Train_acc = 0.99966
Test_acc = 0.98833

The best performing accuracies and settings for BP-CNN is :

Best performing parameters and output is given by:

Optimizer: Adam
Activation function: ReLu
Dropout: 0.7
Learning rate: 0.001
Kernel initializer: RandomUniform
Bias initializer: RandomUniform
Training Loss: 0.003408039682253108
Testing Loss: 0.030509147875125563
Training accuracy: 0.9991333333333333
Testing accuracy: 0.9928

Conclusion: Around 97% of the accuracies are same. 3% are different.

The Back propagation based CNN performs better than FF-CNN by a very small margin in terms of the classification accuracy.

The ensemble model is more robust in architecture, but the FF-CNN classifier has a weak classifier. The model has pseudo labels which are indentified and classified unlike the BP based CNN which optimizes automatically.

Proposal: Weakly supervised classifier for the identification of the classifier is essential to identify the pseudo labels in the fully connected layers of the FF-CNN More data with vast inter-class variance has to be trained to the ensemble model with hard category.

1.c.4 Discussion

From the table obtained from the above, it can be inferred that:
1. The Ensemble model is more robust in its structure since it makes use of different baseline classifier for obtaining the final accuracy
2. The training and the testing accuracies of the Feed forward Convolutional Neural Network has increased in case of the ensemble model.
3. The increase in the accuracy is due to the addition of an ensemble classifier (Support Vector Machine in this case) since Feed forward convolutional neural network has a very weak classifier
4. Diversity is included in the system since, it makes use of different convolutional layer filter sizes (strategy 1 as mentioned in the paper) and to use diversifier input representations (strategy 3 as mentioned in the paper)
5. From the above table, the convolutional layer spatial and the spectral filter size is varied and the data representation is varied by using laws filter to the input images while training the network.
6. Different optimization techiniques of the SVM classifier may obtain better results (linear, polynomial or radial functions)
7. As diversity is included in the system by adopting the two strategies (1 and 3)of the classifier the ensemble model performs better as the system is capable of handling the data with more robust architecture.