Enhancing Convolutional Neural Network using Hu's Moments

Sanad AbuRass[1], Ammar Huneiti[2], Mohammad Belal Al-Zoubi[3]
Computer Science Department, University of Jordan
Amman, Jordan

***Abstract*—Convolutional Neural Networks (CNN) is a powerful deep learning method which is mostly used in image classification and image recognition applications. It has achieved acceptable accuracy in these fields but it still suffers some limitations. One of these limitations of CNN is the lack of ability to be invariant to the input data due to some transformations such as rotation, scaling, skewness, etc. In this paper we present an approach to optimize CNN in order to enhance its performance regarding the invariant limitation by using Hu's moments. The Hu's moments of an image are weighted averages of the image's intensities of the pixels, which produce statistics about the image, and these moments are invariant to image transformations. This means that, even if some changes were made to the image, it will always produce almost the same moments values. The main idea behind the proposed approach is extracting Hu's moments of the image and concatenating them with the flatten vector then feeding the new vector to the fully connected layer. The experimental results show that an acceptable loss, accuracy, precision, recall and F1 score have been achieved on three benchmark datasets which are MNIST hand written digits dataset, MNIST fashion dataset and the CIFAR10 dataset.**

Keywords*—CNN; image transformations; invariant; Hu's moments

I. Introduction

Convolutional Neural Networks (CNN) have achieved an acceptable accuracy in classifying images, but it still suffers some limitations [1],[17],[21]. One of these limitations is the lack of ability to be spatially invariant to the input data due to some transformations [14]. Most present approaches usually use dataset augmentation to solve this issue [2],[4],[21], but this needs larger number of model parameters and more training data, and may result in significantly increased training time and larger chance of under- or overfitting [14],[25]. The effect of this issue is even more obvious when dealing with domain-specific problems. E.g. in medical imaging datasets, the rotation can be extraneous due to the symmetric nature of some biological assemblies. However, the scale is constant during imaging process and should not be deemed as a nuisance factor. Moreover, scale-invariance can decrease the performance if object size is informative, for example, in case of classifying healthy cells from cancer cells [15],[28].

Equivariance and invariance are sometimes used interchangeably but these terms are different from each other. "Equivariance" means varying in a similar or "equivalent proportion" while "invariant" means "no variance at all" [6]. More formally, a function f is equivariant with respect to transformation T if $f(T(x)) = T(f(x))$. This means that, applying the transformation to x is similarly equivalent to applying the transformation to the result $f(x)$. Invariant is a special case of equivariant. A function f is invariant with respect to a transformation T if $f(T(x)) = f(x)$. this means the result through f does not change when a transformation is applied to the input image [27],[8].

CNN is translation equivariance by nature because of the convolution operations [32], since it convolves all over the input image in order to detect the image's features. So, even if an object was shifted, it will still be detected regardless to its position in the image. Also, pooling operations can make CNN rotation equivariance but only if the object was rotated slightly, but as the degree of rotation increases the CNN may fail to classify the object correctly. Although CNN is translation and slight rotation equivariance, it is not translation, scaling or rotation invariant [5],[7],[24],[26],[32],[31].

The problem of transformation invariant in image classification might cause issues in some fields like robotics and autonomous cars. Because of the movement of the robot or the car, the received images might be distorted, translated, scaled or rotated. Therefore, even if the robot or the car is trained to recognize an object they might fail to do so and might cause problems [17],[3],[35]. Image classification is important in surveillance systems to detect unusual activities. Therefore, the invariant problem might cause problems either by classifying an object to be a threat while it is not then making a false alarm, or by classifying an object as a safe object while it is a threat and, in this case, it might lead to a breach of the system [17]. In health care, image classification is used to classify medical images of the patient in order to help diagnose him/her based on the classified images. The invariant problem in this application might cause issues that lead to a misdiagnose of the patient's condition.

The main objective and contribution of this work is to enhance CNN regarding the invariant limitation in order to achieve higher accuracy in image classification by using Hu's moments of the image [23]. The Hu's moments of an image are weighted averages of the image's intensities of the pixels, which produce statistics about the image, and these moments are invariant to image transformations [18], [36]. This means that, even if some changes were made to the image or if the shape outline got slightly thicker, it will always produce almost the same moments values [9]. Therefore, the Hu's moments of the image can be fed to the CNN in order to make it invariant to image transformations. The taxonomy of this

paper will be as follows: Section 2 shows some previous works in this field, Section 3 explains the basics of CNN, Section 4 presents the proposed approach, Section 5 shows the experimental results, and Section 6 shows conclusion.

II. Related Work

Mahesh et al., [18],[19] and Tahmasbi et al. [29] proposed approaches to solve the invariant problem of CNN using Zernike moments. Mahesh et al., [18],[19] proposed a technique which uses Zernike moments in CNN to evaluate the discrimination between face and non-face patterns, and gender classification using facial expression recognition. Their main contribution is the use of Zernike moments as an initial filter, in order to show some unique features of the image that might be helpful to distinguish faces from non-faces image, and gender classifications. They have achieved an accuracy of 100% in distinguishing faces from non-faces images but that is not impressive as it sounds because the discrimination between faces and non-faces is not a hard problem in computer vision any more [10]. In facial expression recognition, they have achieved an accuracy of 87.22%. The main drawback of their work is feature loss. The use of a filter based on Zernike moments might lead to feature loss in some cases.

McNeely-White, et al., [21] Anselmi, et al. [2] and Bruna & Mallat, [4] studied the CNN representations invariance and equivariance to input image transformations. McNeely-White, et al. [21] estimated the linear relationships between representations of the original and transformed images. Although they have achieved good results but their work is considered as data augmentation, and it is not a solid solution to the invariant problem of the CNN.

Cohen & Welling, [7] Gens & Domingos [11] and Mallat [20], analyzed the behavior of the linear representations in relation to symmetry groups, resulting in feature maps that are more invariant to these symmetry groups. Cohen & Welling [7], have revealed that the entire class of such models can be understood mathematically. Although, they have proven their concept mathematically, but their approach still suffers asymmetric world that we live in as described in their own words, “Our approach should also deal better with (approximately) symmetric objects, for which it is not possible to unambiguously estimate pose and motion (what is the pose of a circle?).”. Also, their current model is not suitable for dealing with large images and they consider it as a proof of a concept.

Jaderberg, et al., [14] Hinton [13] and Tieleman [30], have introduced a self-contained module for neural networks. Jaderberg, et al., [14] performed spatial transformations of features by using localization network, parametrized sampling grid, and spatial transformer networks. As in [21] they have used data augmentation which, as it has previously mentioned, is not a robust solution for the invariant problem.

Hinton, et al. [13] and Hinton, et al., [26] proposed a novel CNN architecture which is built up of capsules. These capsules contain group of neurons that are responsible of the instantiation parameters of an entity such as pose velocity and albedo; these capsules will then represent information in a hierarchal from.

The basic theory of their work is that every entity is made up of several smaller entities, so each capsule will try to predict the output of the higher layer capsules, and the capsules which have a greater agreement with the higher layer will be coupled to the parent even more through a positive feedback loop.

Although this work is impressive but it has some shortcomings. The authors have not stated how the weights “W” are learned. Also, the algorithm produces an additional hyper parameter “r” which means more computational complexity. Although the algorithm has achieved the state-of-the-art accuracy on MNIST dataset but it fails to preform so well in CIFAR-10 dataset.

Cheng, et al., [5] Girshick, et al. [12] and Zhang, et al., [34] proposed a method to make CNN rotation invariant. Cheng, et al., [5] added a rotation-invariant layer and Fisher discriminative layer to the CNN in order to make it rotation invariant. These layers will try to learn the objects rotations based on the class, so it can predict the rotation of an object when it recognizes it. They have implemented their algorithm to some famous CNN like VGG and AlexNet, and achieved high accuracy but their work is only directed to rotation invariant, but they did not solve translation or scaling invariant problem in CNN.

Laptev, et al., [15] Su, et al. [28] and Wu, et al., [33] proposed a framework to combine a previous knowledge on nuisance variations with data when training the network. Laptev, et al. [15], formulated a set of transformations and generated multiple images based on these transformations. Then these transformed images are passed through initial layers of the network, and through TI-POOLING operator to from transformation-invariant features. Although they have achieved transformation invariance by pooling transformed features maps, but it added huge computational complexity to the network because of the forward and backward passes for each element.

Worrall, et al., [32] Vedaldi [16] and Memisevic & Hinton, [22] presented a CNN which is equivariant to patch-wise shifting and continuous 360° rotation. Worrall, et al., [32] reconstructed the regular CNN filters by using derivations from complex harmonics, returning a maximal response and orientation for every receptive field patch. Using these derived filters CNN can be invariant to translation and rotation but not scaling. Also, their work has a disadvantage of the higher per-filter computational cost as they must derive and reconstruct all the filters in the CNN.

Up to our knowledge, most of the researches that were studied in the literature review solved the invariant problem of the CNN partially or used data augmentation. In this work, we proposed a general approach to solve the problem with no data augmentation.

III. Convolutional Neural Network

Convolutional Neural Network (CNN) [17] is a major in deep learning which is mostly used in image classification and image recognition tasks due to its convolutional architecture.

Generally, CNN consists of the following phases:

Phase 1: Feature extraction

In this phase, number of filters or kernels will be used to scan the input image, in order to extract features from that image, for example, vertical edges, horizontal edges, corners, etc.

Phase 2: Non-linearity activation

After scanning the filters on the input image, each filter will produce an image which contains the extracted features. The output image must go through a mathematical function which is called Activation Function. In this work, the activation function that will be used is ReLU, which stands for Rectified Linear Unit, which simply converts all the negative values to 0 and keeps the positive values the same as shown in equation (1).

$$R(x) = \text{Max}(0, x) \quad (1)$$

Phase 3: Pooling

Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction.

Phase 4: Dropout

Dropout is used to reduce the CNN overfitting by randomly turning off neurons, so the CNN can take different paths in the training phase. An n-layer fully-connected neural network (ignoring bias) can be defined as:

$$f(x; \{W_i\}_{i \in \{1,\ldots,n\}}) = \Phi_n (W_n \Phi_{n-1} (W_{n-1} \ldots (W_{n-1\ldots} (\Phi_1 (W_1 x)) \quad (2)$$

Phase 5: Input vector extraction

In this phase, CNN converts the 2D matrix to 1D vector, so it can be fed into the neural network.

Phase 6: Network training using the fully connected Neural Network.

Fully connected layer is a neural network which is used to provide the final classification of an image based on matrix mutilation operations, weights and biases. The input of this phase is the flatten vector which was extracted in the previous phase and the output is the predicted classification. CNN can have several fully connected layers where the output of each layer is the input of the next fully connected layer. The objective of a fully connected layer is to take the results of the convolution/pooling process and use them to classify the image into a label. The output of convolution/pooling is flattened into a single vector of values; each of which represents a probability that a certain feature belongs to a label. For example, if the image is of a cat, features representing things like whiskers or fur should have high probabilities for the label "cat". Fig. 1 shows an example of how flatten network is fed to the fully connected layer.

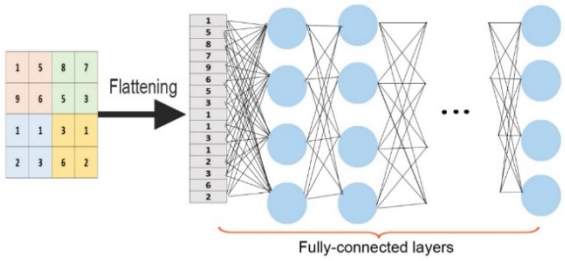

Fig 1. Fully Connected Layer.

IV. Proposed Approach

The main objective of our work is to make CNN invariant to image transformations, in order to achieve higher accuracy in image classification by using Hu's moments of images [23]. The Hu's moments of an image are weighted averages of the image's intensities of the pixels, which produce statistics about the image, and these moments are invariant to image transformations [18], [36]. This means that, even if some changes were made to the image, it will always produce almost the same moments values. Therefore, the Hu's moments of the image can be fed to the fully connected neural network in order to enhance CNN regarding invariant to image transformations limitation.

The invariant features can be achieved using central moments, which are defined as follows [23], [36]:

$$\mu_{pq} = \iint_{-\infty}^{\infty} (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy \quad (3)$$

Where p,q = 0,1,2,…. , $\bar{x} = \frac{m_{10}}{m_{00}}$ and $\bar{y} = \frac{m_{01}}{m_{00}}$

The pixel point $(\bar{x}, \bar{y})$ are the centroid of the image f (x, y). The centroid moments μ_{pq} computed using the centroid of the image f (x, y) is equivalent to the m_{pq} whose center has been shifted to centroid of the image. Therefore, the central moments are invariant to image translations. Scale invariance can be obtained by normalization [36].

The normalized central moments are defined as follows:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \quad \gamma = \frac{p+q+2}{2}, p + q = 2,3, \ldots. \quad (4)$$

Based on normalized central moments,[2,3] introduced seven moment invariants:

$$\emptyset_1 = \boldsymbol{\eta}_{20} + \boldsymbol{\eta}_{02} \quad (5)$$

$$\emptyset_2 = (\boldsymbol{\eta}_{20} - \boldsymbol{\eta}_{02})^2 + 4\boldsymbol{\eta}_{11}^2 \quad (6)$$

$$\emptyset_3 = (\boldsymbol{\eta}_{30} - \mathbf{3}\boldsymbol{\eta}_{12})^2 + (\mathbf{3}\boldsymbol{\eta}_{21} - \boldsymbol{\mu}_{03})^2 \quad (7)$$

$$\emptyset_4 = (\boldsymbol{\eta}_{30} + \boldsymbol{\eta}_{12})^2 + (\boldsymbol{\eta}_{21} + \boldsymbol{\mu}_{03})^2 \quad (8)$$

$$\emptyset_5 = (\boldsymbol{\eta}_{30} - \mathbf{3}\boldsymbol{\eta}_{12})(\boldsymbol{\eta}_{30} + \boldsymbol{\eta}_{12})[(\boldsymbol{\eta}_{30} + \boldsymbol{\eta}_{12})^2 - 3(\boldsymbol{\eta}_{21} + \boldsymbol{\eta}_{03})^2] + (\mathbf{3}\boldsymbol{\eta}_{21} - \boldsymbol{\eta}_{03})(\boldsymbol{\eta}_{21} + \boldsymbol{\eta}_{03})[3(\boldsymbol{\eta}_{30} + \boldsymbol{\eta}_{12})^2 - (\boldsymbol{\eta}_{21} + \boldsymbol{\eta}_{03})^2] \quad (9)$$

$$\emptyset_6 = (\boldsymbol{\eta}_{20} - \boldsymbol{\eta}_{02})[(\boldsymbol{\eta}_{30} + \boldsymbol{\eta}_{12})^2 - (\boldsymbol{\eta}_{21} + \boldsymbol{\eta}_{03})^2] + \mathbf{4}\boldsymbol{\eta}_{11}(\boldsymbol{\eta}_{30} + \boldsymbol{\eta}_{12})(\boldsymbol{\eta}_{21} + \boldsymbol{\eta}_{03}) \quad (10)$$

$$\emptyset_{7=}(3\eta_{21}-\eta_{03})(\eta_{30}+\eta_{12})[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2]-(\eta_{30}-3\eta_{12})(\eta_{21}+\eta_{03})[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2] \quad (11)$$

The adopted research methodology comprises the following steps, as shown in Fig. 2:

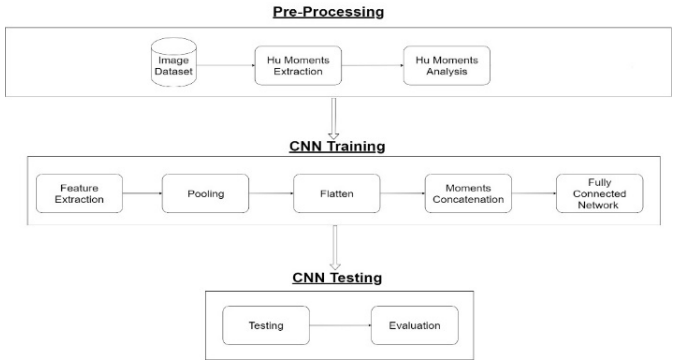

Fig 2. Methodology Phases.

After the flattening operation, Hu's moments of the original image are concatenated with the flattened vector, so the CNN can recognize these values in the testing phase.

Hu's moments concatenation should make the flatten vector more informative and expository. The new vector will be fed to the fully connected network; therefore, CNN will be trained to see the Hu's moments, extent and solidity values alongside with the features vector, these values will affect the neurons' activations in the network in order to achieve transformation invariant. Fig. 3 shows an example of Hu's moments concatenation.

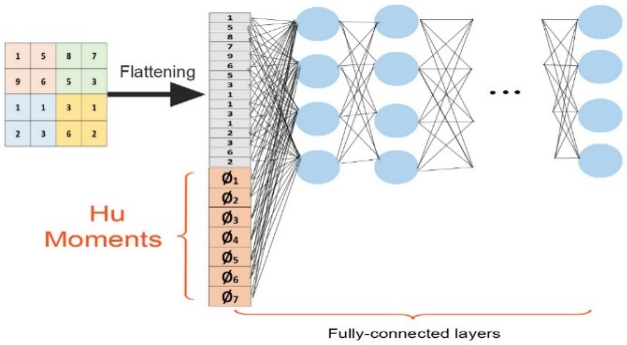

Fig 3. Hu's Moments Concatenation.

V. Experimental Results

We have implemented the proposed approach using Python TensorFlow platform powered by Google Colab notebook. We have tested our approach on three datasets which were MNIST handwritten digits dataset, MNIST fashion dataset and CIFAR10 dataset. Finally, we have compared our results with the work of [18] which uses Zernike moments (ZM) as an initial filter to extract invariant features of the images, as motioned above, by implementing their approach on the three datasets. ZM are projections of an image on to the complex Zernike polynomials that are orthogonal over the unit circle. So, a radius must be provided in order to calculate the ZM of the image. Therefore, we have used the degrees 45° and 90° to extract ZMs of the images. Our approach has archived better loss, accuracy, precision, recall and F1 score compared with the work of [18] on the three dataset MNIST hand written digits, MNIST fashion dataset and CIFAR 10 dataset. The use of Zenick moments as initial filters led to feature loss which led to a decrease in loss, accuracy, precision, recall and F1 score. On the other hand, adding Hu's moments to the flattening vector led to discriminative and more informative vector therefore a better performance.

Table I and Table II shows the results of our approach compared to the results of [18] approach on MNIST handwritten digits dataset. Fig. 4, Fig. 5 and Fig. 6 illustrate the loss, accuracy precision, recall and F1 Score respectively and they show that our approach achieved better performance than [18] approach.

TABLE I. MNIST Handwritten Digits Loss and Accuracy Comparisons

Approach	*# of Epochs*	*Loss (cross-entropy)*	*Accuracy*
Original CNN	30	0.062	97.1%
	50	0.022	98.21%
	100	0.018	98.1%
(Mahesh et al. 2017) 45 Degree	30	0.042	97.7%
	50	0.002	98.81%
	100	0.014	98.4%
(Mahesh et al. 2017) 90 Degree	30	0.03	97.8%
	50	0.031	98.1%
	100	0.004	98.4%
Our approach	30	0.004	98.8%
	50	0.006	99%
	100	0.002	99.2%

TABLE II. MNIST Handwritten Digits Precision, Recall and F1 Score Comparisons

Approach	*# of Epochs*	*Precision*	*Recall*	*F1 Score*
Original CNN	30	99.22%	99.42%	99.61%
	50	99.72%	99.56%	99.73%
	100	99.81%	99.78%	99.8%
(Mahesh et al. 2017) 45 Degree	30	99.83%	99.72%	99.82%
	50	99.88%	99.77%	99.85%
	100	99.93%	99.82%	99.88%
(Mahesh et al. 2017) 90 Degree	30	99.85%	99.75%	99.84%
	50	99.90%	99.81%	99.87%
	100	99.92%	99.85%	99.89%
Our approach	30	99.94%	99.85%	99.9%
	50	99.95%	99.88%	99.92%
	100	99.96%	99.91%	99.94%

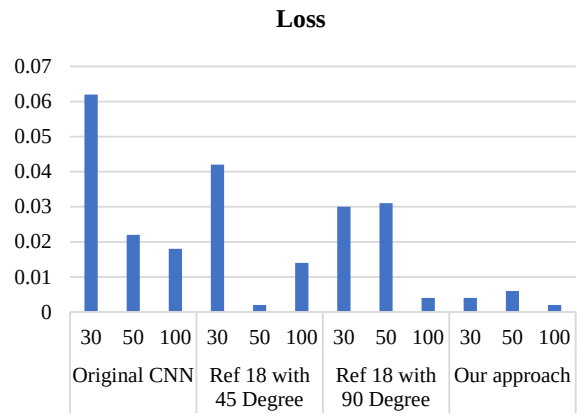

Fig 4. MNIST Handwritten Digits Loss Comparison.

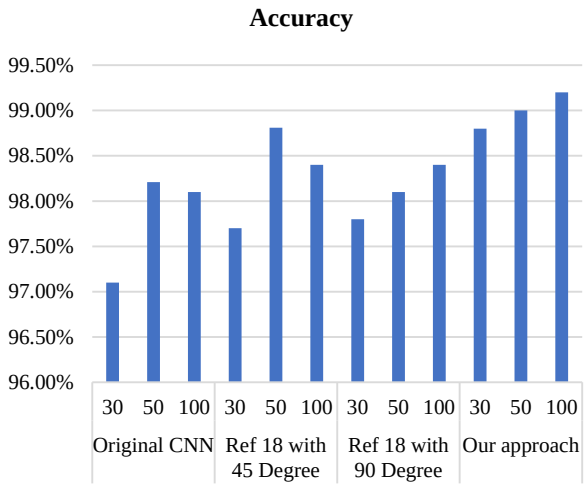

Fig 5. MNIST Handwritten Digits Accuracy Comparison.

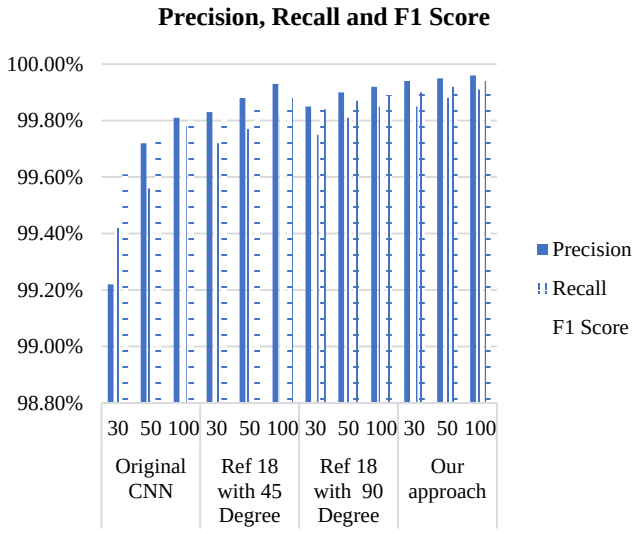

Fig 6. MNIST Handwritten Digits Precision, Recall and F1 Score Comparison.

Table III and Table IV and Fig. 7, 8 and 9 below show the results of our approach implemented on MNIST fashion dataset and we have compared our work with [18] approach and we achieved better results compared to their work.

TABLE III. MNIST FASHION LOSS AND ACCURACY COMPARISONS

Approach	*# of Epochs*	*Loss (cross-entropy)*	*Accuracy*
Original CNN	30	0.35	83.9%
	50	0.235	84.8%
	100	0.249	86.2%
(Mahesh et al. 2017) 45 Degree	30	0.25	84.6%
	50	0.216	85.9%
	100	0.228	87.1%
(Mahesh et al. 2017) 90 Degree	30	0.268	84.4%
	50	0.224	85.7%
	100	0.176	86.8%
Our approach	30	0.251	89.3%
	50	0.076	90.05%
	100	0.024	91.7%

TABLE IV. MNIST FASHION PRECISION, RECALL AND F1 SCORE COMPARISONS

Approach	*# of Epochs*	*Precision*	*Recall*	*F1 Score*
Original CNN	30	97.24%	97.9%	97.3%
	50	97.69%	98.04%	97.82%
	100	97.88%	98.19%	98.07%
(Mahesh et al. 2017) 45 Degree	30	97.74%	98.03%	97.88%
	50	97.9%	98.14%	98.02%
	100	97.99%	98.3%	98.15%
(Mahesh et al. 2017) 90 Degree	30	97.82%	97.85%	97.84%
	50	97.94%	98.01%	97.97%
	100	98.01%	98.21%	98.11%
Our approach	30	98.51%	97.85%	98.18%
	50	98.55%	98.14%	98.34%
	100	98.67%	98.28%	98.47%

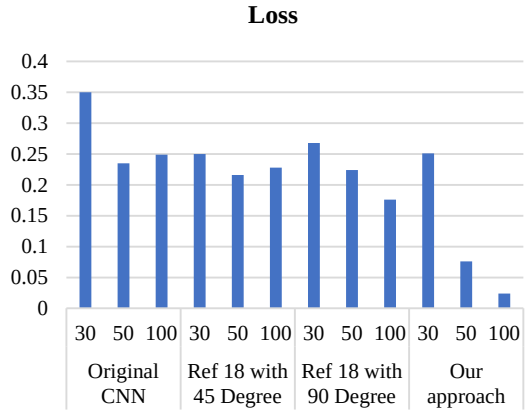

Fig 7. MNIST Fashion Loss Comparison.

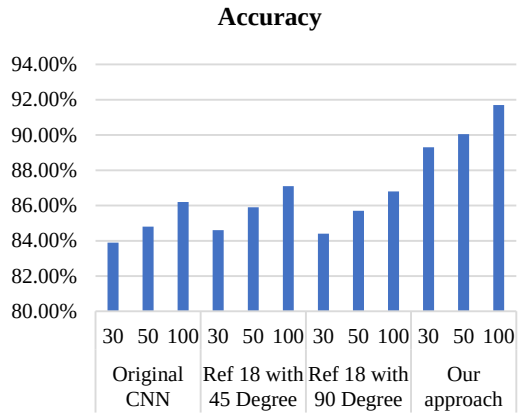

Fig 8. MNIST Fashion Accuracy Comparison.

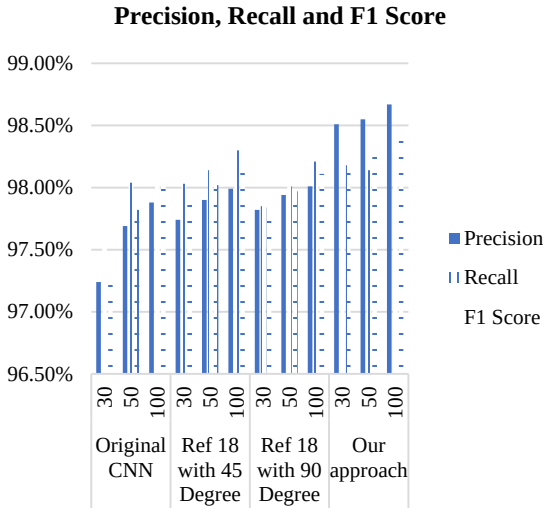

Fig 9. MNIST Fashion Precision, Recall and F1 Score Comparison.

Finally, we have tested our approach and [18] approach on CIFAR10 dataset and we have achieved better performance than their approach as shown below in Table V and Table VI and Fig. 10, 11 and 12.

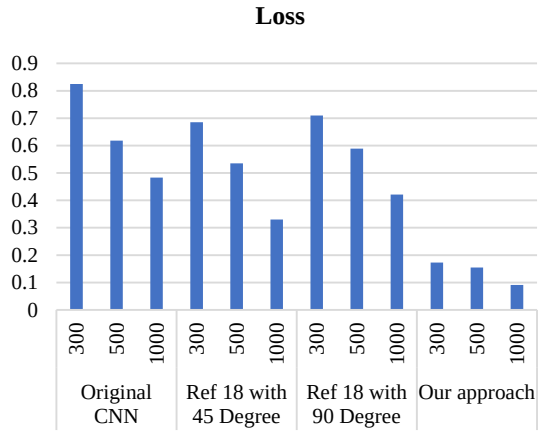

Fig 10. CIFAR10 Loss Comparison.

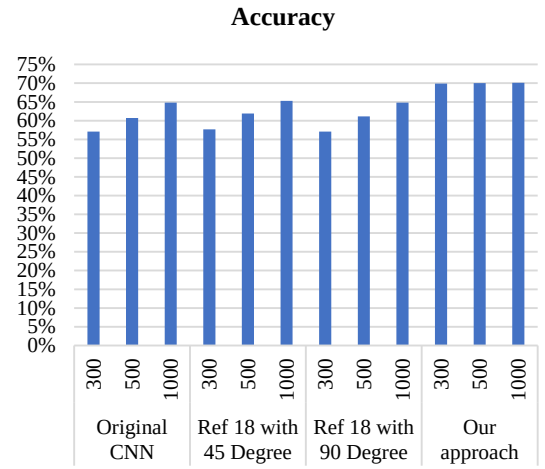

Fig 11. CIFAR10 Accuracy Comparison.

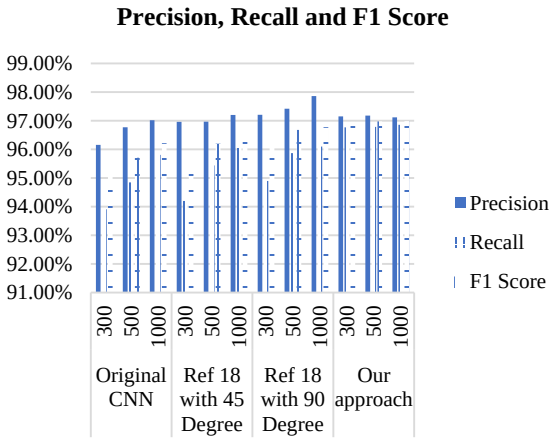

Fig 12. CIFAR10 Precision, Recall and F1 Score Comparison.

TABLE V. CIFAR10 LOSS AND ACCURACY COMPARISONS

Approach	*# of Epochs*	*Loss (cross-entropy)*	*Accuracy*
Original CNN	300	0.825	57.1%
	500	0.618	60.69%
	1000	0.483	64.83%
(Mahesh et al. 2017) 45 Degree	300	0.685	57.7%
	500	0.535	61.9%
	1000	0.33	65.3%
(Mahesh et al. 2017) 90 Degree	300	0.710	57.1%
	500	0.589	61.12%
	1000	0.421	64.8%
Our approach	300	0.173	69.9%
	500	0.155	70%
	1000	0.091	70.1%

TABLE VI. CIFAR10 PRECISION, RECALL AND F1 SCORE COMPARISONS

Approach	*# of Epochs*	*Precision*	*Recall*	*F1 Score*
Original CNN	300	96.16%	93.91%	95.06%
	500	96.77%	94.85%	95.71%
	1000	97.02%	95.81%	96.22%
(Mahesh et al. 2017) 45 Degree	300	96.96%	94.2%	95.56%
	500	96.97%	95.44%	96.2%
	1000	97.2%	96.05%	96.62%
(Mahesh et al. 2017) 90 Degree	300	97.21%	94.9%	96.11%
	500	97.42%	95.87%	96.68%
	1000	97.86%	96.1%	96.78%
Our approach	300	97.15%	96.77%	96.96%
	500	97.18%	96.79%	96.98%
	1000	97.12%	96.86%	96.99%

Fig. 13, 14 and 15 show some real prediction after implementing our approach on MNIST Hand Written Digits Dataset, MNIST Fashion Dataset and CIFAR10 Dataset, respectively.

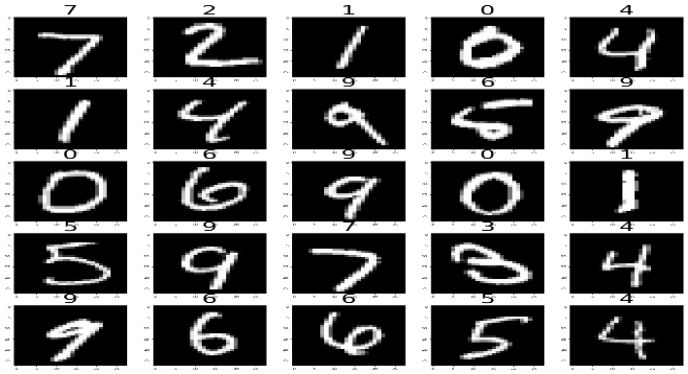

Fig 13. Results of MNIST Hand Written Digits Classification.

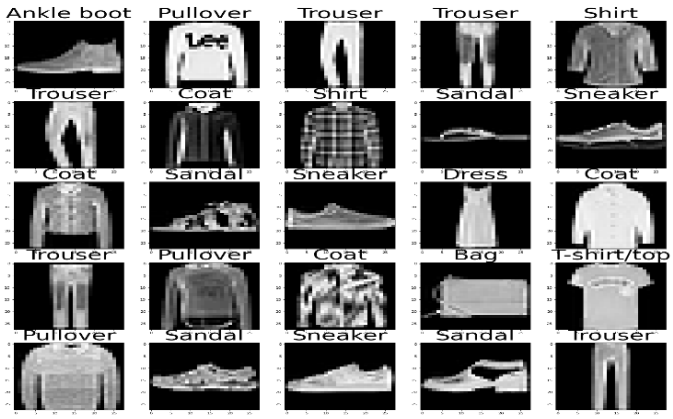

Fig 14. Results of MNIST Fashion Dataset Classification.

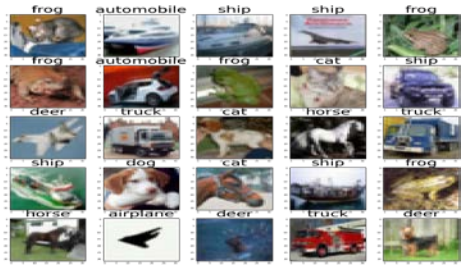

Fig 15. Results of CIFAR10 Dataset Classification.

VI. DISCUSSION

CNN suffers from the problem of being invariant to image transformations. Up to our knowledge most previous researches solved this problem partially or they used data augmentation. Our approach uses Hu's moments to make the flatten vector more descriptive so when it is fed to the fully connected layer it should lead to a better classification regardless to the image transformations. Concatenating the moments with the flatten vector was challenging, since the vector size will increase and it should be the same as the input size of the fully connected layer.

VII. CONCLUSION

This paper presents an approach to enhance CNN regarding the invariant problem by using Hu's moments. The mechanism behind this approach is by concatenating Hu's moments with the flattening vector before feeding it to the fully connected layer in order to make the vector more discriminative and more informative. In this study we have implemented our approach then we have compared out work with the work of Mahesh et al., on the three dataset MNIST hand written digits, MNIST fashion dataset and CIFAR 10 dataset. The results show that our method gave best results in all cases namely loss, accuracy, precision, recall and F1 score. The main limitation of our work is the fixed sizes of the flatten vector that means the size of the vector should precalculated and predefined so it be the same as the size of the input size of the fully connected layer.

REFERENCES

[1] M. Z. Alom et al., "A State-of-the-Art Survey on Deep Learning Theory and Architectures," Electronics, vol. 8, no. 3, p. 292, Mar. 2019, Accessed: Oct. 07, 2020. [Online].

[2] F. Anselmi, J. Z. Leibo, L. Rosasco, J. Mutch, A. Tacchetti, and T. Poggio, "Unsupervised learning of invariant representations," Theor. Comput. Sci., vol. 633, pp. 112–121, Jun. 2016.

[3] M. Billinghurst, A. Clark, and G. Lee, "A survey of augmented reality," 2015, [Online]. Available: http://ir.canterbury.ac.nz/handle/10092/15494.

[4] J. Bruna and S. Mallat, "Invariant scattering convolution networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1872–1886, Aug. 2013.

[5] G. Cheng, J. Han, P. Zhou, and D. Xu, "Learning Rotation-Invariant and Fisher Discriminative Convolutional Neural Networks for Object Detection," IEEE Trans. Image Process., vol. 28, no. 1, pp. 265–278, Jan.2019.

[6] B. Chidester, M. N. Do, and J. Ma, "Rotation Equivariance and Invariance in Convolutional Neural Networks," arXiv [stat.ML], May 31, 2018.

[7] T. Cohen and M. Welling, "Group Equivariant Convolutional Networks," in International Conference on Machine Learning, Jun. 2016, pp. 2990–2999, Accessed:Oct.07,2020.[Online].

[8] T. S. Cohen and M. Welling, “Transformation Properties of Learned Visual Representations,” arXiv [cs.LG],Dec.24,2014.

[9] J. Flusser, T. Suk, and B. Zitová, “3D moment invariants to translation, rotation, and scaling,” in 2D and 3D Image Analysis by Moments, Wiley, 2016, p. 96.

[10] K. S. Gautam and T. Senthil Kumar, “Discrimination and Detection of Face and Non-face Using Multilayer Feedforward Perceptron,” in Proceedings of the International Conference on Soft Computing Systems, 2016, pp. 89–103.

[11] R. Gens and P. M. Domingos, “Deep Symmetry Networks,” in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2537–2545.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[13] G. E. Hinton, A. Krizhevsky, and S. D. Wang, “Transforming Auto-Encoders,” in Artificial Neural Networks and Machine Learning – ICANN 2011, 2011, pp. 44–51.

[14] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial Transformer Networks,” in Advances in Neural Information Processing Systems 28, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2017–2025.

[15] D. Laptev, N. Savinov, J. M. Buhmann, and M. Pollefeys, “TI-POOLING: transformation-invariant pooling for feature learning in convolutional neural networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 289–297.

[16] K. Lenc and A. Vedaldi, “Learning Covariant Feature Detectors,” in Computer Vision – ECCV 2016 Workshops, 2016, pp. 100–117.

[17] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, “A survey of deep neural network architectures and their applications,” Neurocomputing, vol. 234, pp. 11–26, Apr. 2017.

[18] V. G. V. Mahesh, A. N. J. Raj, and Z. Fan, “Invariant moments based convolutional neural networks for image analysis,” International Journal of Computational Intelligence Systems, vol.10, no.1, pp. 936–950, 2017.

[19] V. G. V. Mahesh and A. N. J. Raj, “Invariant face recognition using Zernike moments combined with feed forward neural network,” Int. J. Biom., vol. 7, no. 3, pp. 286–307, Jan. 2015.

[20] S. Mallat, “Group Invariant Scattering,” Commun. Pure Appl. Math., vol. 65, no. 10, pp. 1331–1398, Oct. 2012.

[21] D. G. McNeely-White, J. Ross Beveridge, and B. A. Draper, “Inception and ResNet: Same Training, Same Features,” Advances in Intelligent Systems and Computing. pp. 352–357, 2020, doi: 10.1007/978-3-030-25719-4_45.

[22] R. Memisevic and G. E. Hinton, “Learning to represent spatial transformations with factored higher-order Boltzmann machines,” Neural Comput., vol. 22, no. 6, pp. 1473–1492, Jun. 2010.

[23] Ming-Kuei Hu, “Visual pattern recognition by moment invariants,” IRE Transactions on Information Theory, vol.8, no.2, pp.179–187,Feb. 1962.

[24] Y. Poleg, A. Ephrat, S. Peleg, and C. Arora, “Compact CNN for indexing egocentric videos,” in 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), Mar. 2016, pp. 1–9.

[25] I. Rocco, R. Arandjelovic, and J. Sivic, “Convolutional neural network architecture for geometric matching,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 6148–6157.

[26] S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic Routing Between Capsules,” in Advances in Neural Information Processing Systems 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 3856–3866.

[27] R. Serfling, “Equivariance and invariance properties of multivariate quantile and related functions, and the role of standardisation,” J. Nonparametr. Stat., vol. 22, no. 7, pp. 915–936, Oct. 2010.

[28] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, “Multi-view convolutional neural networks for 3d shape recognition,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 945–953.

[29] A. Tahmasbi, F. Saki, and S. B. Shokouhi, “Classification of benign and malignant masses based on Zernike moments,” Comput. Biol. Med., vol. 41, no. 8, pp. 726–735, Aug. 2011.

[30] T. Tieleman, Optimizing neural networks that generate images. University of Toronto (Canada), 2014.

[31] S. Vosoughi, P. Vijayaraghavan, and D. Roy, “Tweet2Vec: Learning Tweet Embeddings Using Character-level CNN-LSTM Encoder-Decoder,” in Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, Pisa, Italy, Jul. 2016, pp. 1041–1044, Accessed: Oct. 06, 2020. [Online].

[32] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, “Harmonic networks: Deep translation and rotation equivariance,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5028–5037.

[33] J. Wu, Y. Yu, C. Huang, and K. Yu, “Deep multiple instance learning for image classification and auto-annotation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3460–3469.

[34] Y. Zhang, K. Sohn, R. Villegas, G. Pan, and H. Lee, “Improving object detection with deep convolutional networks via bayesian optimization and structured prediction,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 249–258.

[35] Z. Zhang and M. Sabuncu, “Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels,” in Advances in Neural Information Processing Systems 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 8778–8788.

[36] Zhihu Huang and Jinsong Leng, “Analysis of Hu’s moment invariants on image scaling and rotation,” in 2010 2nd International Conference on Computer Engineering and Technology, Apr. 2010, vol. 7, pp. V7–476–V7–480.

[37] Z. Zhu et al., “A Configurable Multi-Precision CNN Computing Framework Based on Single Bit RRAM,” in 2019 56th ACM/IEEE Design Automation Conference (DAC), Jun. 2019, pp. 1–6.

AUTHORS’ PROFILE

Sanad AbuRass received the B.S. degree and M.Sc. degree in computer science from Al Balqa Applied University, As-Salt, Jordan in 2013 and 2016, respectively. From 2015 to 2020 he was ICT Teacher/ IT Administrator at AHSS school, where he worked on school’s management systems. Currently, he is a Ph.D. candidate in computer science at University of Jordan, Amman Jordan. His current interests include image processing, computer vision and deep learning.

Ammar M. Huneiti received his BSc, MSc and PhD degrees from Cardiff University, UK. His BSc is in Computer Science, his MSc is in Information Systems Technologies and his PhD is in Systems Engineering. Between 1992 and 2000 he worked for several private and public sector organizations supervising the design and implementation of IT related projects. In addition, he served as a senior consultant to the ministry of Social Development in Jordan and the National Aid Fund. At present, he is a Full Professor at the Department of Computer Information Systems, King Abdullah II School of Information Technology, the University of Jordan. His research interests include, Intelligent Information Systems, Machine Learning, Spatial Data Mining, and Image Classification.

Mohammad Belal Al-Zoubi is a Professor of Machine Learning and Digital Image Processing in the Department of Computer Information Systems at the Universityof Jordan. He received a B.S. in Cybernetics from The University of Bllgrade in 1985. Prof. Al-Zoubi received his M.S. in Information Systems for the University of Detroit, 1985, and his PhD in Computer Science from Leeds University, UK, 1995. (e-mail: mba@ju.edu.j