IJEMD-CSAI, Special Issue (2026) https://doi.org/10.54938/ijemdcsai.2026.04.2.610

International Journal of Emerging Multidisciplinaries:
Computer Science and Artificial Intelligence

Research Paper
Journal Homepage: www.ojs.ijemd.com
ISSN (print): 2791-0164 ISSN (online): 2957-5036

Convolutional Neural Networks for Multi-Class Image Classification:

A Custom CNN Approach on Fashion-MNIST

Andrew Leewei Hobbs[1] , Abdul Salam Shah Sayed[1]

1. Department of Computer Science, Taylor's University, Malaysia

Abstract

The given work is a design, implementation and evaluation of a tailored Convolutional Neural Network (CNN) that can be trained to provide multi-class image classification in ten different clothing categories based on the Fashion-MNIST benchmark data set [13]. The data sets include 70,000 28x28 grayscale 28x28 pixel images (60,000 training data and 10,000 test data). In the proposed CNN architecture, three convolutional-pooling blocks each containing a filter depth of 32, 64 and 128 are used, then finally a fully connected classifier with a softmax output layer to generate class probabilities in the ten categories. The data preprocessing steps involved pixel value rescaling in [0, 1] range, reshaping tensors to meet the Conv2D layer requirements, stratified train/validation/test splitting, and sparse integer label encoding with the sparse categorical cross-entropy loss. The Adam optimizer [7] was used, with validation-based callbacks, i.e. Early Stopping and ReduceLROnPlateau, to reduce overfitting and guarantee generalizable behavior. After testing on the held-out test set, the final model had test accuracy of about 92% and test loss of about 23 percent indicating high generalization to unknown data. The confusion analysis has shown that classification errors were clustered across the visually similar category of upper-body garments, in particular, shirt, T-shirt, coat, and pullover, which is also very common in the Fashion-MNIST literature and is also primarily due to the low pixel density of the dataset [13]. This conclusion is supported by the fact that a small,

purposely designed CNN architecture is an effective and computationally efficient solution to this benchmark classification problem, and that the performance gaps still exist largely due to natural limits of the data sets, and not due to architecture weaknesses.

Keywords: Convolutional Neural Network (CNN), softmax, Fashion-MNIST

INTRODUCTION

Image classification Image classification Image classification A problem in computer vision and deep learning that is one of the most fundamental and commonly studied is the problem of assigning a semantic label to an input image, based on its visual content. Its significance goes far beyond the boundaries of academic standards: autonomous vehicles perception, satellite imagery, content moderation, e-commerce products classifications, and more all heavily rely on the capacity to process raw visual inputs into structured and actionable information [1]. Although it has been developed over several decades, image classification is still a non-trivial, because of such factors as intra-class variation (when an object of the same type looks visually different because of its pose, lighting or texture), inter-class similarity (visually similar objects of different categories), background clutter, and partial occlusion.

Before the advent of deep learning, the prevailing paradigm in image classification was based on a two stage pipeline: manual feature extraction, and a conventional supervised classifier. To encode the local structural and gradient features in fixed-length features, algorithms like Scale-Invariant Feature transform (SIFT) [2] and Histogram of Oriented Gradients (HOG) [3] were created and then transformed into the inputs of the classifier which included Support Vector Machines (SVMs) [4], or k-Nearest Neighbours (k-NN). Although the pipelines have performed competitively on constrained metrics, they were fragile and domain-specific in the sense that a model trained on pedestrian detection does not need to be retrained on clothing recognition without substantial modification.

The development of deep Convolutional Neural Networks (CNNs) radically changed the picture in image classification. Instead of using fixed feature descriptors, CNNs are trained to learn a hierarchical representation using the raw pixel data, by end-to-end gradient-based optimisation [5]. The initial CNN layers learn low-level primitives, such as edges, corners and simple textures, but the deeper layers of the CNN learn to combine these primitives into increasingly abstract representations,

that are class-discriminative. AlexNet [6] was a particularly influential demonstration of this hierarchical feature learning paradigm: it achieved a new state-of-the-art top-5 error rate on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and sparked a frenzy of transition to deep learning in virtually all subfields of computer vision.

In this paper, an inquiry into the use of a small, specially developed CNN on the Fashion-MNIST dataset [7], a more challenging drop-in replacement of the initial MNIST handwritten digit dataset, is made. Fashion-MNIST was created specifically to reveal the weaknesses of the simple classifiers with a more realistic distribution of visual patterns, and has found extensive use as a benchmark to compare classification architectures. The selected data has certain challenges, such as low image resolution (28x28 pixels), lack of colour detail (grayscale), and existence of various visually ambiguous pairs of classes the most noticeable of which exists among upper-body garments, such as shirts, T-shirts, pullovers, and coats.

This paper does not follow the general trend of adapting a big-scale pre-trained network, but rather explicitly builds a small, understandable CNN using first principles. This design decision is inspired by pedagogical transparency, computational efficiency, and compatibility with a dataset: large models like VGG-16 [8] or ResNet-50 [9], which are trained on high-resolution RGB input images, would add unnecessary parameter overhead and risk overfitting when used without significant modification on 28x28 grayscale input images. The given architecture follows the principles of the established CNN design, such as local receptive fields, weight sharing, and hierarchical pooling, yet it is small enough to be trained at high speeds and using limited resources[10].

The rest of this paper is organized as follows: Section 2 summarizes the existing literature on the topic in the field of image classification, both the classical approaches based on features and the current CNN models; Section 3 has a description of the dataset and all preprocessing steps that were applied; Section 4 is the description of the model architecture and training configuration; Section 5 presents and discusses the experimental results; and finally, Section 6 is the conclusion with a summary of results and the direction of further improvement.

LITERATURE REVIEW

1.Classical Feature Engineering Methods.

Prior to the rise of deep learning as the new paradigm, the study of image classification had been focused on the construction of hand-designed feature descriptors that were both capable of capturing these discriminative visual patterns and resistant to the common transformations of an image[11-12]. The SIFT and HOG were two of the most influential descriptors to be created during this period.

SIFT is a method that finds local keypoints multiple scales and orientations, and describes each keypoint with a 128-dimensional gradient orientation histogram. This architecture makes SIFT features independent of uniform scale and rotation, and somewhat resistant to illumination variations properties which contributed to its great efficiency in matching and retrieval. SIFT descriptors however are computationally costly to compute and a separate matching / classification step would be required, and are not as well adapted to end-to-end learning pipelines[13-14].

HOG, performs the task of pedestrian detection, breaks down an image into a grid of very small cells and calculates a normalised histogram of gradient orientations in each of the cells. Those local histograms are further concatenated to create a global feature that describes how the edge and shape information is distributed across the image. HOG was very successful in rigid object categories with clear outline structure, but performs badly on deformable objects or ones with a large intra-class variance which Fashion-MNIST shares with clothing items.

SIFT or HOG extracted descriptors were generally combined with classifiers like SVMs or k -nn. SVMs, find a maximum-margin separating hyperplane between classes in an input representation of controllable dimension, and were once known for their ease of use at the price of scalability and sensitivity to feature quality and choice of distance measure. k-NN assigns labels based on the majority class of the k training samples nearest in the feature space to the query, which is simple at the cost of scalability and sensitivity to feature quality and choice of distance measure. Both classifiers have a severe reliance on the quality of the input features: in case the feature that captures the

appropriate discriminative structure is not captured by the descriptors, neither of the classifiers are able to reconstruct meaningful class boundaries.

This reliance on hand-crafted characteristics forms the key shortcoming of the classical pipeline. The feature engineers have to know domain specific information and invest huge amount of effort into describing and validating descriptors to each new job and thus the classical pipelines are costly to generalise and hard to adapt to new datasets with different visual properties[15-16].

2. Convolutional Neural Networks and Learned Representations

[9] provided a conceptual framework of CNNs by building upon LeNet-5 a multi-layer network of convolutional and pooling stages followed by fully connected layers and trained by backpropagation to perform handwritten digit recognition. LeNet-5 showed that weight-sharing filters with a local connection were capable of learning spatially significant features using raw pixel as inputs, and set the basic CNN architecture of extracting features succeeded by a classifier that is still widely used today[17-19].

The scalability of CNNs was finally demonstrated conclusively with AlexNet, which scored a 5th place on ILSVRC with a 15.3% top-5 error rate, and its nearest non-CNN competitor scored over ten percentage points worse. AlexNet introduced a number of new architectural and training techniques that became the new norm: Rectified Linear Unit (ReLU) activations, used to speed up training and combat vanishing gradients; dropout regularisation, used to prevent co-adaptation of feature detectors; local response normalisation; and data augmentation. More importantly, AlexNet was trained using Graphics Processing Units (GPUs) which has allowed the parallelisation needed to support a network of this size.

Following architectures followed systematic depth scaling as a roadmap towards performance enhancement. It was shown by VGGNet [11] that the receptive fields of large-kernel convolutions can be matched with a sequence of smaller 3x3 filters with fewer parameters and higher non-linearity per parameter count. VGG-16 and VGG-19 set state-of-the-art results on ILSVRC 2014, and were popular feature extractors due to their simple uniform architecture in transfer learning pipelines.

[5] recognized a fundamental barrier to training very deep networks: higher the network depth, the smaller the exponentially decreasing gradients passing through all the layers, and the higher the performance level or the worse the performance level vanishing gradient problem. Their solution, Residual Networks (ResNets), proposed identity skip connections, which cut through one or more layers and enabled identity gradients to pass through the skip connection and enabled it to train networks with hundreds of layers. ResNet-50 and ResNet-152 have new state-of-the-art results on ILSVRC 2015, and residual connections have become an almost universal part of state-of-the-art design.

3.Fashion-MNIST, benchmarking

[13] introduced Fashion-MNIST as a more difficult alternative to the original MNIST digit dataset, which had already become critiqued as too easy to be a significant benchmark that can be achieved by a simple linear classifier with an over 90 percent accuracy rate. Fashion-MNIST uses a format (28x28 grayscale images, 10 classes, 70,000 samples) identical to that of MNIST, but they replace handwritten digits with ten classes of fashion items (T-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot).

The dataset is a significantly more challenging set to classify than MNIST especially in the upper-body garment classes. A linear classifier only brokers about 83.9% the accuracy of Fashion-MNIST as compared to 92.6% on MNIST, and a CNN with two convolutional layers fends about 91.6% demonstrating the more challenging nature of the task and the specific benefit CNNs possess in this instance of a spatial classification task [13]. The benchmark has since been used extensively to test new architectures, regularisation plans and data augmentation strategies in the research community.

METHODOLOGY

1. Dataset

The experimental benchmark in this research was Fashion-MNIST dataset [13]. The data consists of 70,000 uniformly distributed 28x28 pixel greyscale images, which are an equal number of images in ten mutually exclusive clothing categories T-shirt/top (0), Trouser (1), Pullover (2), Dress (3), Coat

(4), Sandal (5), Shirt (6), Sneaker (7), Bag (8), and Ankle Boot (9). The official partition of datasets provides 60,000 training images and 10,000 testing images.

Other benchmarks were avoided because fashion-MNIST has a greater inter-class visual similarity and it learns spatial features in a more natural way, but can be trained with compact custom architecture due to its computational simplicity. The small size of the dataset (28x28 pixels, single channel) and the clear organization of classes allow attending to a small image, which is very suitable to illustrating the main principles of hierarchical convolutional learning of features, which is the main goal of the given research.

2. Data Preprocessing

2.1 Pixel Value Rescaling

The Fashion-MNIST data is represented as raw pixel values using unsigned 8-bit integers of the range 0 through 255, which is the intensity of greyscale. Before training, pixel values were normalised to [0, 1] by first changing them to 32-bit floating point and then dividing it by 255.0. This normalisation step is common in deep learning pipelines [4], and serves two useful functions: it makes the magnitude of gradients at backpropagation less sensitive[20-22] to the absolute magnitude of the input values, which facilitates more stable optimisation; and it causes faster convergence by matching the distributions of input values with the typical operating range of common activation functions like ReLU.

2.2 Tensor Reshaping

Conv2D Keras layers take input tensors in the form (batch size, height, width, channels). Since the images of Fashion-MNIST are in grey scale, every image is first represented as a 2D array of size (28, 28). To meet the Conv2D input shape (28, 28, 1), every image Tensor was reshaped to (28, 28, 1) categorically using the singleton channel dimension[23-25]. This is needed to properly perform the spatial convolution otherwise TensorFlow/Keras will build a model with a shape mismatch error.

2.3 Train / Validation / Test Split

The official test split of 10,000 images was not accessed during any point but was not evaluated until final model evaluation. Out of the official training images of 60,000, 10,000 were isolated as

validation set, and 50,000 images were used to train the model. The validation set plays two purposes (i) it acts as a performance signal to track the generalisation in training and (ii) it is the trigger measure of the Early Stopping and ReduceLROnPlateau callbacks in Section 3.3. All splits were made with a fixed random seed in order to make them reproducible across experimental runs[26-28].

2.4 Label Encoding

The labels of the datasets are represented by integer values [0-9]. Instead of using one-hot encoding to encode each integer label into a 10-dimensional binary representation to be used with categorical cross-entropy loss each integer label was kept as a sparse integer and entered into the sparse_categorical_crossentropy loss function. Sparse categorical cross entropy is mathematically identical to categorical cross-entropy on one-hot encoded labels [4] but does not require the construction of an explicit one-hot matrix, which saves on memory during training and makes the preprocessing pipeline easier[29-31].

3. Model Architecture

A Sequential CNN architecture was built that had three convolutional-pooling blocks, a flatten operation, a fully connected hidden layer with dropout regularisation and a softmax output layer. The architecture has standard CNN design rules in terms of learning spatial features: the first few layers learn low-level primitives; the latter layers learn to combine these primitives into class-discriminative features[32-33]

3.1 Input Layer

The model takes tensors of size (28, 28, 1), which is the height, width, and single greyscale channel of Fashion-MNIST images.

3.2 Convolutional Block 1

The initial block uses the same 32 3x3 filters with same padding and ReLU activation that leave the spatial size of the input intact (28x28) but generates 32 feature maps. This shallow block has a mission of identifying low level features such as edges, corners and plain textures that make up the building blocks of clothing shape recognition. MaxPooling is then applied twice to reduce the spatial resolution to 14x14 to decrease the number of computations and provide some translation invariance [9].

3.3 Convolutional Block 2

The second block uses 64 3x3 filters, with padding same and ReLU activation, as the input of the previous pooling layer 14x14x32 feature maps. This block can be trained to know mid-level combinations of features structures that include collar edges, sleeve shapes, and handle shapes, by increasing the number of filters in the block to 64. The spatial resolution is furthered to 7x7 by a following 2x2 MaxPooling.

3.4 Convolutional Block 3

The third block applies 128 filters of size 3×3 with 'same' padding and ReLU activation to the 7×7×64 feature maps. This deeper block targets class-specific, high-level representations fine-grained textural and structural patterns needed to discriminate between visually similar categories such as shirts and T-shirts. A final 2×2 MaxPooling reduces the spatial resolution to 3×3, yielding a compact 3×3×128 feature volume.

3.5 Flatten and Fully Connected Layers

The volume of the 3x3x128 feature is reduced to a 1,152 dimensional vector. Nonlinear combinations of all extracted features are then learned with a fully connected (Dense) layer having 128 units and ReLU activation to represent an entire category of clothes globally. This Dense layer is followed by a Dropout layer whose rate is 0.5, which deactivates half of the neurons randomly on each training step. Dropout is a common regularisation method that minimises the co-adaptation of feature detectors and is empirically shown to enhance generalisation in deep networks [12].

3.6 Output Layer

The final layer is the Dense layer of 10 units and softmax activation, generating a probability distribution of all Fashion-MNIST classes, which contain 10. This distribution is the predicted class, which is the argmax of the predicted class[34-36]. The architecture contains 241,546 a trainable parameters which is enough to train meaningful feature hierarchies given the 28x28 imagery of Fashion-MNIST, but is small enough to prevent overfitting or incur prohibitively high training time.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 32)	320
max_pooling2d (MaxPooling2D)	(None, 14, 14, 32)	0
conv2d_1 (Conv2D)	(None, 14, 14, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 7, 7, 64)	0
conv2d_2 (Conv2D)	(None, 7, 7, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 3, 3, 128)	0
flatten (Flatten)	(None, 1152)	0
dense (Dense)	(None, 128)	147,584
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 10)	1,290

Total params: 241,546 (943.54 KB)
Trainable params: 241,546 (943.54 KB)
Non-trainable params: 0 (0.00 B)

Figure 1: CNN model architecture summary

4. Training Configuration

This model was collected using the Adam optimiser [7], an adaptive gradient descent algorithm that keeps learning rate estimates per parameter based on first and second order gradient moment estimates. Adam was chosen because it has historically been shown to have better empirical performance compared to normal Stochastic Gradient Descent (SGD) on deep learning problems, especially its resistance to choice of learning rate and rapid convergence to sparse gradients. The original learning rate was 0.001, which is a default recommendation [7]. Two Keras callbacks controlled training. Early Stopping measured validation loss using a patience of 5 epochs and it stopped training and loaded the weights of the lowest validation loss in validation loss observed during the patience. ReduceLROnPlateau has a method of reducing the learning rate by half when the validation loss does not make any progress over 3 consecutive plateau epochs so it can make finer gradient steps in parts of the loss landscape where the coarser updates are stuck at optimisation. Both the calls backs are set standards in training deep networks with a scanty amount of data [4].

EVALUATION AND RESULTS

1.Training and Validation Performance

In training, it was found that both training and validation accuracy improved persistently during the first epochs. The rate of validation accuracy and validation loss reduced exponentially around epoch 6 when the validation metrics started to level off. The accuracy of training and training loss kept on improving after this stage and this is a hint of mild overfitting, which is a typical feature of deep networks during the initial stages where the networks memorise patterns in training and do not generalise with the rest of the data distribution [4]. Early Stopping callback was able to identify this divergence and prevent further training, which restored model weights that were associated with the lowest epoch validation loss. The last checkpoint model had a validation accuracy of about 93% and a validation loss of about 22% which is on par with well-trained CNNs in the Fashion-MNIST literature using similar architectures [13]. The convergence behaviour of the training curves quick initial learning and subsequent slow plateau and early stopping behaviour is indicative of proper model capacity in relation to the complexity of the dataset. A common characteristic of undercapacity models is that they level off at lower values of accuracy and do not reduce further with further training, whereas overcapacity models show increasing differences between training and validation loss. The relatively narrow difference between the training and validation convergence accuracy proves that dropout regularisation effectively eliminated co-adaptation and severe overfitting [12].

2. Final Test Set Evaluation

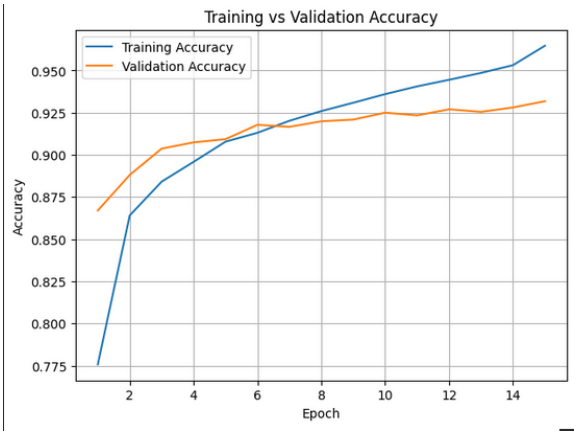

Figure 2: Final test set accuracy and loss scores

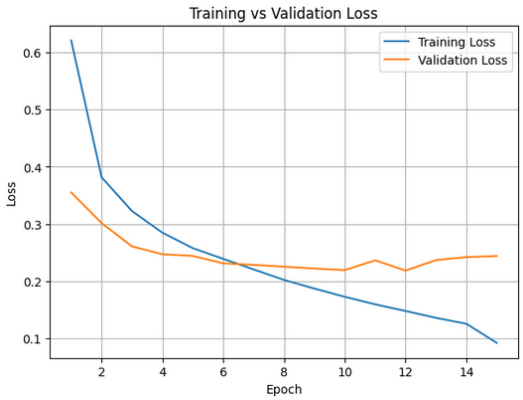

Figure 3: Final test set accuracy and loss scores

When tested on the fully held-out test set (10,000 images), the model had a test accuracy of about 92% and test loss of about 23%. The fact that the marginal reduction of about one percentage point compared with the validation accuracy is natural and reflects the change in the natural distribution between the validation and the test partition; it does not imply that it is overfitting to the validation partition. The fact that validation and test performance are close to each other validates the fact that the model was successfully generalised to unknown data. The per-class classification report analysis indicated that the model has an excellent precision and recall on visually distinct classes particularly Trouser (class 1), Sandal (class 5), Sneaker (class 7) and Bag (class 8) which have strong, unique, class-specific visual features. Conversely, there was a significant decrease in performance in upper-body garment categories, in particular, Shirt (class 6), T-shirt/top (class 0), Coat (class 4), and Pullover (class 2), and the respective classes had a disproportionately high misclassification rate.

3. Confusion Matrix Analysis

The confusion diagram gave detailed information about the organization of model errors. The prevalent off-diagonal entries affirmed that the misclassifications were concentrated in the category topwear cluster: shirts were often mistaken with T-shirts/tops, pullovers were mixed with coats and vice versa. The trend aligns with the results indicated by [13] in the original Fashion-MNIST article and has been replicated in many separate benchmarking studies. This observed pattern of confusion is explained by two complementary ways. First, fashion images represented by Fashion-MNIST images (28x28 pixels) have a low spatial resolution, which limits the amount of high-frequency

textural detail, like fabric weave, neckline detail, and button position, that might otherwise serve as discriminative information to draw a distinction between similar types of garments. Fine-grained features to the point that they are below a representational threshold of any convolutional filter are compressed at this resolution, independent of model capacity. Second, the fact that aggressive spatial downsampling of the feature map of 28x28 to 3x3 across three pooling steps can remove the remaining fine-grained spatial information which might otherwise be useful in disambiguation can be seen as an additional factor contributing to the inaccuracy of the model. In the future, it may be useful to decrease the pooling strides or include dilated convolutions to retain more spatial resolution in deeper layers [5].

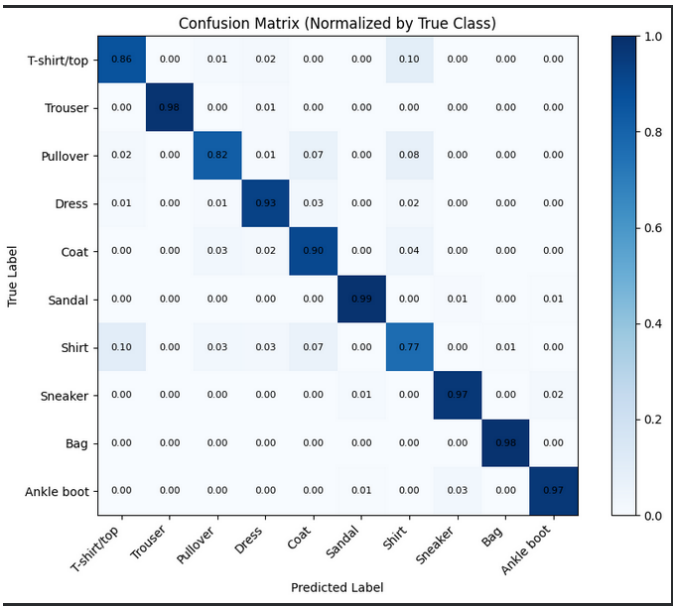

Figure 4: Normalized confusion matrix evaluated on the 10,000-image test set. Diagonal values represent correct classification rates per class; off-diagonal values indicate misclassification proportions.

	precision	recall	f1-score	support
T-shirt/top	0.86	0.86	0.86	1000
Trouser	0.99	0.98	0.99	1000
Pullover	0.91	0.82	0.87	1000
Dress	0.91	0.93	0.92	1000
Coat	0.84	0.90	0.87	1000
Sandal	0.98	0.99	0.98	1000
Shirt	0.76	0.77	0.76	1000
Sneaker	0.96	0.97	0.96	1000
Bag	0.99	0.98	0.99	1000
Ankle boot	0.97	0.97	0.97	1000
accuracy			0.92	10000
macro avg	0.92	0.92	0.92	10000
weighted avg	0.92	0.92	0.92	10000

F1 (macro): 0.9176735988835001
F1 (weighted): 0.9176735988835

Figure 5: Per-class precision, recall, and F1-score on the 10,000-image test set. Overall accuracy: 92%. Macro and weighted F1: 0.918.

4. Misclassification Analysis

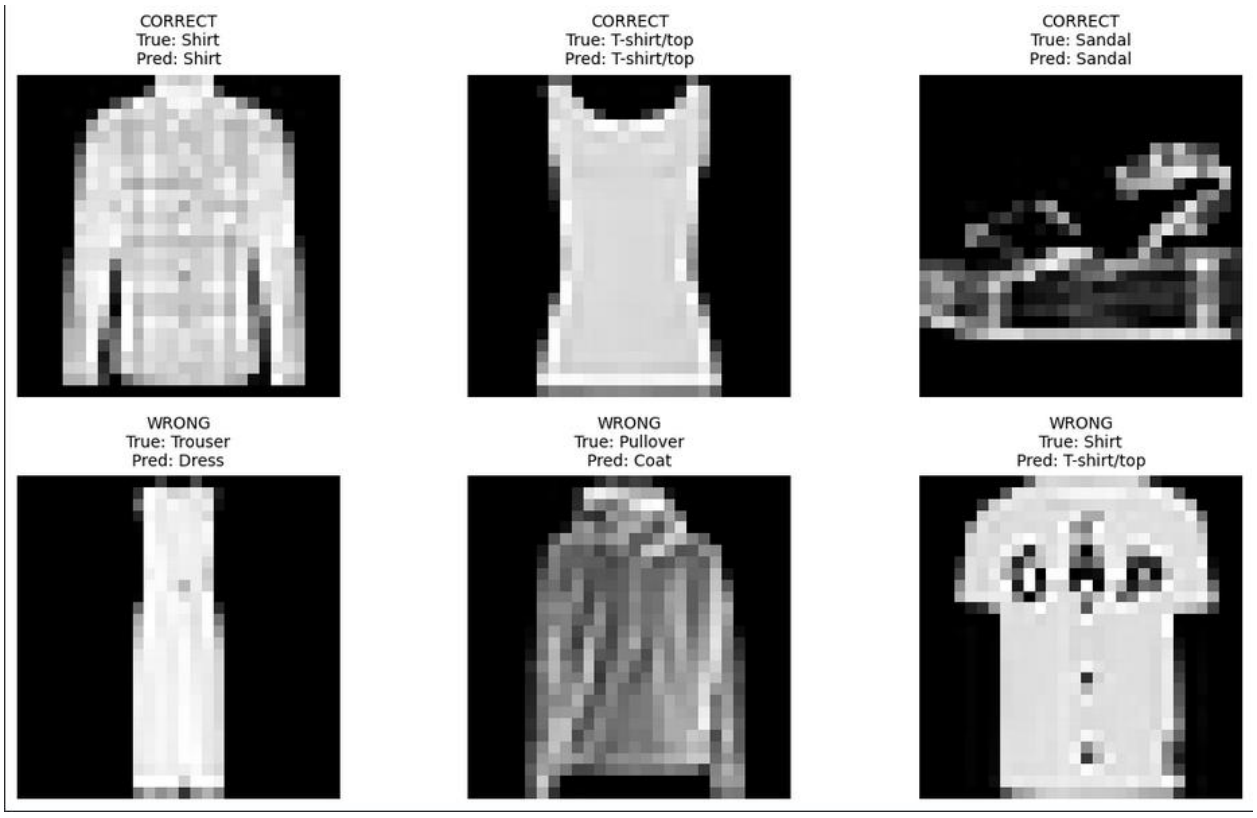

Figure 6: Example misclassified images with true and predicted labels.

This aims at examining the fallacy of misclassification. The qualitative examination of individual misclassified instances proved that the false predictions were not chance occurrences. Within the analyzed examples, the predicted class visually overlapped the true one by a large margin: a pullover

in the case of a coat, at the resolution of 28x28, had a silhouette and body profile, which could not be distinguished against that of a coat, without the details of the fine texture. This result supports the assumption that it is the inherent vagueness of low image resolution that causes the first error and not necessarily some inherent deficiency in the feature learning ability of CNN. It is also observed that the finite capacity of the model due to the spatial resolution of the deepest feature maps of 3x3 is possibly too small to discern fine differences in the fabric texture and cut that are partially coded even in a 28x28 image. Adding more filters to the third convolutional block or cutting the pooling stride, or using separable convolutions [1] are some of the possible improvements to the topwear subgroup without significantly raising the computational cost.

CONCLUSION

The research presented, designed, and tested a small custom Convolutional Neural Network that can classify multi-class images on the Fashion-MNIST benchmark dataset [13]. The proposed architecture three stacked convolutional-pooling blocks whose depths are 32, 64 and 128 and then a fully connected classifier with dropout regularisation and softmax output obtained the test accuracy of about 92% and test loss of about 23% in the held-out test partition, signifying a strong generalisation ability to unknown data. The paper validated some of the established concepts of deep learning in image classification. To begin with, the hierarchical convolutional feature learning starting with low level edge detectors in low level features through to class discriminant structural representations in high level features allows effective spatial pattern extraction with even small 28x28 grey scale inputs [9]. Second, dropout regularisation was effective in curbing overfitting because it prevented co-adaptation of feature detectors in a training run [12]. Third, validation-based training callbacks Early Stopping and ReduceLROnPlateau were useful to find the optimal stopping point and the learning rate schedule, which resulted in a well-fitted model without underfitting and excessive overfitting [4]. Examination of the confusion results and the examples of misclassifications showed that the residual error was not random but being structural around the visually similar classes of upper body garments shirts, T-shirts, coats and pullovers. The pattern of this error is in line with the established challenge of Fashion-MNIST topwear subgroup, which has been reported in the literature on a wide variety of architectures [13]. The first factor is the fact that fine-grained textural and structural detail is lost at 28x28 pixel resolution, and thus discriminatory cues below the representational threshold of convolutional filters are lost irrespective of model depth. The second

reason is the aggressive spatial downsampling of the three MaxPooling steps which reduces the feature map size to 3x3 spatial resolution and has the potential to eliminate any residual discriminative information. The inherent benefit of end-to-end learned representations was verified by comparing them to classical feature-based algorithms such as SIFT [10], HOG [3] and SVM classifiers [2]: CNNs bypass the domain-specific feature engineering bottleneck by learning discriminative features directly to the source using backpropagation to produce more robust and generalisable classifiers. Further development of the work could take many directions to enhance the performance of the Fashion-MNIST format within its limitations. Intra-class variation Light data augmentation such as random horizontal flips, small rotations, and small shear transformations may help enhance intra-class variation robustness. Architectural adaptations like smaller pooling strides, depthwise separable convolutions [1], or batch normalisation [6] can help better feature retention and training stability. Alternatively, the transfer learning of a lightweight backbone might be used to give richer original feature representations without compromising computational tractability. Overall, the present research paper proves that a small purpose-specific CNN model is a viable, interpretable, and computationally efficient method of achieving multi-class image classification on Fashion-MNIST, and that the current level of accuracy is limited by the resolution of datasets and not by architectural factors.

REFERENCES

[1] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 1251–1258). IEEE.

[2] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297. https://doi.org/10.1007/BF00994018

[3] Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) (Vol. 1, pp. 886–893). IEEE. https://doi.org/10.1109/CVPR.2005.177

[4] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press. https://www.deeplearningbook.org

[5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 770–778). IEEE. https://doi.org/10.1109/CVPR.2016.90

[6] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning (ICML) (pp. 448–456). PMLR.

[7] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR). arXiv:1412.6980.

[8] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems (NeurIPS) (Vol. 25, pp. 1097–1105). Curran Associates. https://doi.org/10.1145/3065386

[9] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324. https://doi.org/10.1109/5.726791

[10] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

[11] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556. https://arxiv.org/abs/1409.1556

[12] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929–1958.

[13] Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747. https://arxiv.org/abs/1708.07747

[14] Gordienko, Y., Trochun, Y., & Stirenko, S. (2024). Multimodal quanvolutional and convolutional neural networks for multi-class image classification. *Big Data and Cognitive Computing*, *8*(7), 75.

[15] Murugan, P. (2018). Implementation of deep convolutional neural network in multi-class categorical image classification. *arXiv preprint arXiv:1801.01397*.

[16] Negi, A., Kumar, K., & Chauhan, P. (2021). Deep neural network-based multi-class image classification for plant diseases. *Agricultural informatics: automation using the IoT and machine learning*, 117-129.

[17] Khalil, M.I., Humayun, M., Jhanjhi, N.Z., Talib, M.N., Tabbakh, T.A. (2021). Multi-class Segmentation of Organ at Risk from Abdominal CT Images: A Deep Learning Approach. In: Peng, SL., Hsieh, SY., Gopalakrishnan, S., Duraisamy, B. (eds) Intelligent Computing and Innovation on Data Science. Lecture Notes in Networks and Systems, vol 248. Springer, Singapore. https://doi.org/10.1007/978-981-16-3153-5_45

[18] Murthy, V. N., Singh, V., Chen, T., Manmatha, R., & Comaniciu, D. (2016). Deep decision network for multi-class image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2240-2248).

[19] Zheng, H., Sherazi, S. W. A., Son, S. H., & Lee, J. Y. (2021). A deep convolutional neural network-based multi-class image classification for automatic wafer map failure recognition in semiconductor manufacturing. *Applied Sciences*, *11*(20), 9769.

[20] Sarki, R., Ahmed, K., Wang, H., Zhang, Y., & Wang, K. (2021). Convolutional neural network for multi-class classification of diabetic eye disease. *EAI Endorsed Transactions on Scalable Information Systems*, *9*(4).

[21] Niveshitha, N., Amsaad, F., & Jhanjhi, N. Z. (2023, August). Air quality prediction in smart cities using cloud machine learning. In *2023 Second International Conference On Smart Technologies For Smart Nation (SmartTechCon)* (pp. 1115-1119). IEEE.

[22] Heenaye-Mamode Khan, M., Boodoo-Jahangeer, N., Dullull, W., Nathire, S., Gao, X., Sinha, G. R., & Nagwanshi, K. K. (2021). Multi-class classification of breast cancer abnormalities using Deep Convolutional Neural Network (CNN). *Plos one*, *16*(8), e0256500.

[23] Almazroi, A. A., Alsubaei, F. S., Ayub, N., & Jhanjhi, N. Z. (2024). Inclusive Smart Cities: IoT-Cloud Solutions for Enhanced Energy Analytics and Safety. *International Journal of Advanced Computer Science & Applications*, *15*(5).

[24] Ninama, H., Raikwal, J., Ravuri, A., Sukheja, D., Bhoi, S. K., Jhanjhi, N. Z., ... & Abdelmaboud, A. (2024). Computer vision and deep transfer learning for automatic gauge reading detection. *Scientific Reports*, *14*(1), 23019.

[25] Nawaz, M., Sewissy, A. A., & Soliman, T. H. A. (2018). Multi-class breast cancer classification using deep learning convolutional neural network. *Int. J. Adv. Comput. Sci. Appl*, *9*(6), 316-332.

[26] Rashmi, S., Siwach, V., Sehrawat, H., Brar, G. S., Singla, J., Jhanjhi, N. Z., ... & Shorfuzzaman, M. (2024). AI-powered VM selection: Amplifying cloud performance with dragonfly algorithm. *Heliyon*, *10*(19).

[27] Anwesa Chaudhuri, A. C., & Sanjib Ray, S. R. (2015). Antiproliferative activity of phytochemicals present in aerial parts aqueous extract of Ampelocissus latifolia (Roxb.) Planch. on apical meristem cells.

[28] Youneszade, N., Marjani, M., & Ray, S. K. (2023). A predictive model to detect cervical diseases using convolutional neural network algorithms and digital colposcopy images. *IEEE Access*, *11*, 59882-59898.

[29] Mehmood, A., Maqsood, M., Bashir, M., & Shuyuan, Y. (2020). A deep Siamese convolution neural network for multi-class classification of Alzheimer disease. *Brain sciences*, *10*(2), 84.

[30] Talo, M., Yildirim, O., Baloglu, U. B., Aydin, G., & Acharya, U. R. (2019). Convolutional neural networks for multi-class brain disease detection using MRI images. *Computerized Medical Imaging and Graphics*, *78*, 101673.

[31] Ezat, W. A., Dessouky, M. M., & Ismail, N. A. (2020, January). Multi-class image classification using deep learning algorithm. In *Journal of Physics: Conference Series* (Vol. 1447, No. 1, p. 012021). IOP Publishing.

[32] Zhang, J., Zhang, Q., Ren, J., Zhao, Y., & Liu, J. (2022, May). Spatial-context-aware deep neural network for multi-class image classification. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1960-1964). IEEE.

[33] S. M. Muzammal, R. K. Murugesan, N. Z. Jhanjhi and L. T. Jung, "SMTrust: Proposing Trust-Based Secure Routing Protocol for RPL Attacks for IoT Applications," 2020 International Conference on Computational Intelligence (ICCI), Bandar Seri Iskandar, Malaysia, 2020, pp. 305-310, doi: 10.1109/ICCI51257.2020.9247818.

[34] Chaturvedi, S. S., Tembhurne, J. V., & Diwan, T. (2020). A multi-class skin Cancer classification using deep convolutional neural networks. *Multimedia Tools and Applications*, *79*(39), 28477-28498.

[35] Brohi, S. N., Jhanjhi, N. Z., Brohi, N. N., & Brohi, M. N. (2020). Key Applications of State-of-the-Art Technologies to Mitigate and Eliminate COVID-19.pdf.. https://doi.org/10.36227/techrxiv.12115596.v1

[36] Wang, Y., Deng, Y., Zheng, Y., Chattopadhyay, P., & Wang, L. (2025). Vision transformers for image classification: A comparative survey. *Technologies*, *13*(1), 32.