mathematics

MDPI

Article

State-of-the-Art Results with the Fashion-MNIST Dataset

Ravil I. Mukhamediev [1,2]

[1] Institute of Automation and Information Technologies, Satbayev University (KazNRTU), 22 Satpayev Street, Almaty 050013, Kazakhstan; r.mukhamediev@satbayev.university
[2] Institute of Information and Computational Technologies, CS MSHE RK (Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan), 28 Shevchenko Street, Almaty 050010, Kazakhstan

Abstract: In September 2024, the Fashion-MNIST dataset will be 7 years old. Proposed as a replacement for the well-known MNIST dataset, it continues to be used to evaluate machine learning model architectures. This paper describes new results achieved with the Fashion-MNIST dataset using classical machine learning models and a relatively simple convolutional network. We present the state-of-the-art results obtained using the CNN-3-128 convolutional network and data augmentation. The developed CNN-3-128 model containing three convolutional layers achieved an accuracy of 99.65% in the Fashion-MNIST test image set. In addition, this paper presents the results of computational experiments demonstrating the dependence between the number of adjustable parameters of the convolutional network and the maximum acceptable classification quality, which allows us to optimise the computational cost of model training.

Keywords: Fashion-MNIST; convolution neural networks; accuracy

MSC: 68T01; 68T05; 68T07; 68Q32; 97P80

check for updates

Citation: Mukhamediev, R.I. State-of-the-Art Results with the Fashion-MNIST Dataset. *Mathematics* **2024**, *12*, 3174. https://doi.org/10.3390/math12203174

Academic Editor: Daniel-Loan Cuiiac

Received: 4 August 2024
Revised: 19 September 2024
Accepted: 3 October 2024
Published: 11 October 2024

1. Introduction

Datasets play an important role in the development of increasingly advanced machine learning models. This paper discusses the possibility of improving the classification of garments presented in the Fashion-MNIST dataset. The task of image classification is one of the most frequently studied computer vision problems. To solve the computer vision problems and to test classifiers, the MNIST dataset was developed in the 1990s [1]. This dataset presents handwritten digits, each in the form of a black and white picture of 28×28 pixels in size. The set contains 60,000 images for training the machine learning models and 10,000 images for quality assessment. The mentioned paper [1] presents in chronological order the results achieved by different machine learning models using this famous dataset. However, over time, the researchers have improved the performance of digit recognition, which is comparable to that of humans and can be even better [2]. There was a need to update the dataset to provide researchers with new possibilities for model development, hyperparameter tuning, and computational experiments. For this purpose, the Fashion-MNIST dataset, containing clothing images, was proposed in 2017. This dataset is posited as more complex and fully replicates the MNIST structure: 70,000 black and white images of 28×28 pixels, of which, as before, 60,000 images are used for training and 10,000 images are used for model quality assessment. The authors presented the results of a large number of computational experiments [3] using 14 models with different hyperparameter values. The best result reported in the literature is demonstrated in [4], where the model called cnn-dropout-3 achieved 99.1% accuracy.

This work aims to achieve a better result compared to the one obtained in [4], and to evaluate how the number of adjustable convolutional network parameters can limit the maximum achievable model quality.

2. Related Works

As is mentioned above, the Fashion-MNIST dataset has the same structure as the original MNIST dataset. Since Fashion-MNIST has appeared, it has been used as a test dataset for testing various machine learning models [5] and for solving some practical problems in classifying clothing images [6,7], extracting the clothing data from the images [8]. It should be noted that there are other datasets for classifying clothing items: [9] (1893 images), [10] (800 images), [11] (80,000 images), [12] Adidas AG ™, DeepFashion [13] (300,000 images), and [14] (more than 1.2 million images). However, due to its similarity to its famous predecessor, Fashion-MNIST continues to attract researchers interested in computer vision tasks. The following results achieved by machine learning models using Fashion-MNIST are mentioned in the literature (see Table 1):

Table 1. Some results achieved with the Fashion-MNIST dataset.

Model	Accuracy	Reference
Boosted Trees (GBM/XGBoost)	85.3	[15]
DecisionTreeClassifier	79.8	[3]
ExtraTreeClassifier	77.5	[3]
GaussianNB	51.1	[3]
KNeighborsClassifier	85.4	[3]
Linear support vector classificator (SVC)	83.6	[3]
LogisticRegression	84.2	[3]
MLPClassifier	87.1	[3]
RandomForestClassifier	87.3	[3]
SVC	89.7	[3]
Long short-term memory (LSTM)	88.26	[5]
Extreme learning machines (ELMs)	97	[16]
Two-layer convolution neural network (CNN) along with Batch Normalization and Skip Connections	92.54	[17]
CNN	93	[18]
CNN	93.43	[19]
Shallow CNN	93.59	[20]
CNN4 + HPO + Reg	93.99	[21]
VGG16 H-CNN	93.52	[22]
VGG19 H-CNN	93.33	[22]
CNN using Adam	94.52	[23]
CNN LeNet-5	98	[24]
CNN-dropout-3	99.1	[4]
CNN-2-128 with image augmentation	99.65	This article

As the table shows, the best results are achieved using the convolutional neural networks (CNNs) as expected. For this reason, we will also use this model to achieve the best results.

3. Fashion-MNIST Dataset

Although the composition of the Fashion-MNIST dataset is well known, we will give a brief description of it here. The set consists of 70,000 black and white 28×28 images of items of clothing labelled as follows: 0: T-shirt/top; 1: Trouser; 2: Pullover; 3: Dress; 4: Coat;

5: Sandal; 6: Shirt; 7: Sneaker; 8: Bag; 9: Ankle boot. Examples of these image are shown in the Figure 1.

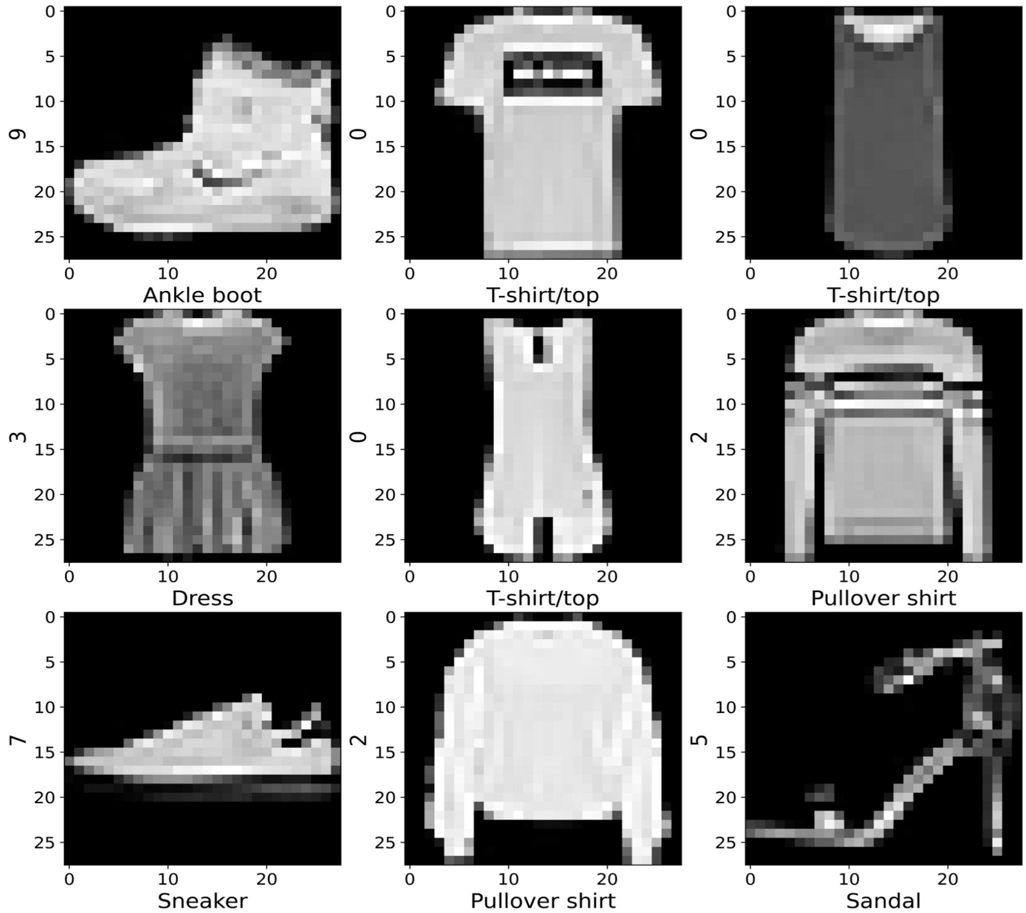

Figure 1. Some clothing items from the Fashion-MNIST dataset.

In total,10,000 images are used as the test set, and 60,000 are used as the training set. The dataset is balanced; in other words, it has an equal number of clothing items of different classes.

4. Methods

The method consists of the following steps:

1. Loading the dataset;
2. Standard preprocessing in the form of normalisation and dimensionality transformation to input in the convolutional network;
3. Connecting the augmenter and configuring it accordingly;
4. Training and evaluating the model results.

Figure 2 shows the main steps in the computational experiments.

The dataset is loaded with one line of code:

(x_train, y_train), (x_test, y_test) = keras.datasets.fashion_mnist.load_data()

Preprocessing consists of transforming the data dimensionality and normalization.

Points 3 and 4 are iterative in order to find the best parameters of the augmentation model and the convolution network model. We used the in-house developed image augmentation software and the more advanced Image Data Generator as augmenters at different stages of the computational experiments. Image Data Generator is a class of TensorFlow library that is used to generate tensor images with the temporal augmentation of the number of images. It allows for augmenting the images directly during the model training.

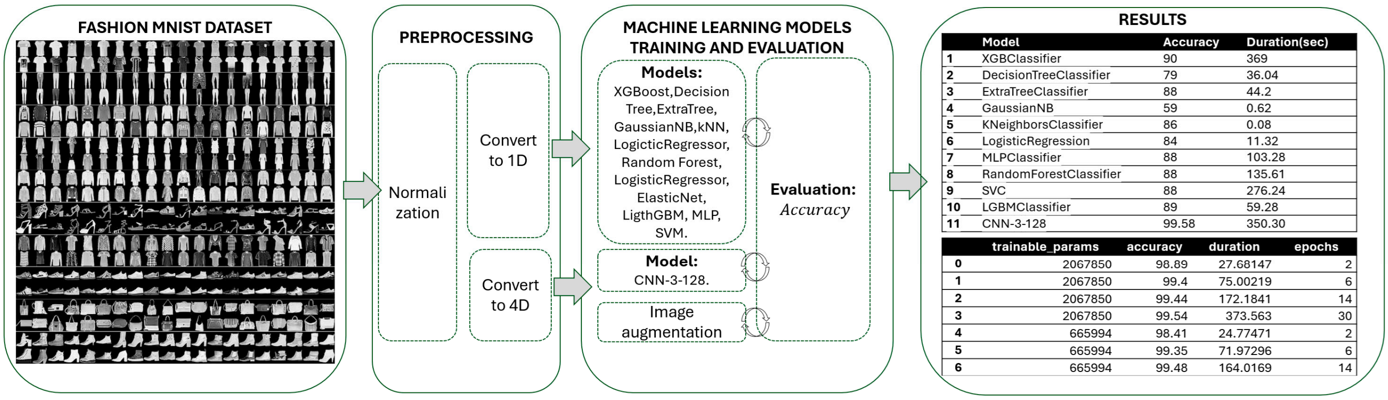

Figure 2. Main stages of computational experiments.

The main parameters of Image Data Generator are given in Appendix A. In the process of implementing the experiments, we used almost the same set of machine learning models as in the above-mentioned article [3].

However, to achieve the best results, we used a convolutional network of relatively simple architecture, including three convolutional layers, dropout layers, and two forward connection layers with the softmax function at the network output. The parameters of the layers are shown in Figure 3. The convolutional layers are of the same type, with a dimension of 3×3 and Max Pooling with a dimension of 2×2.

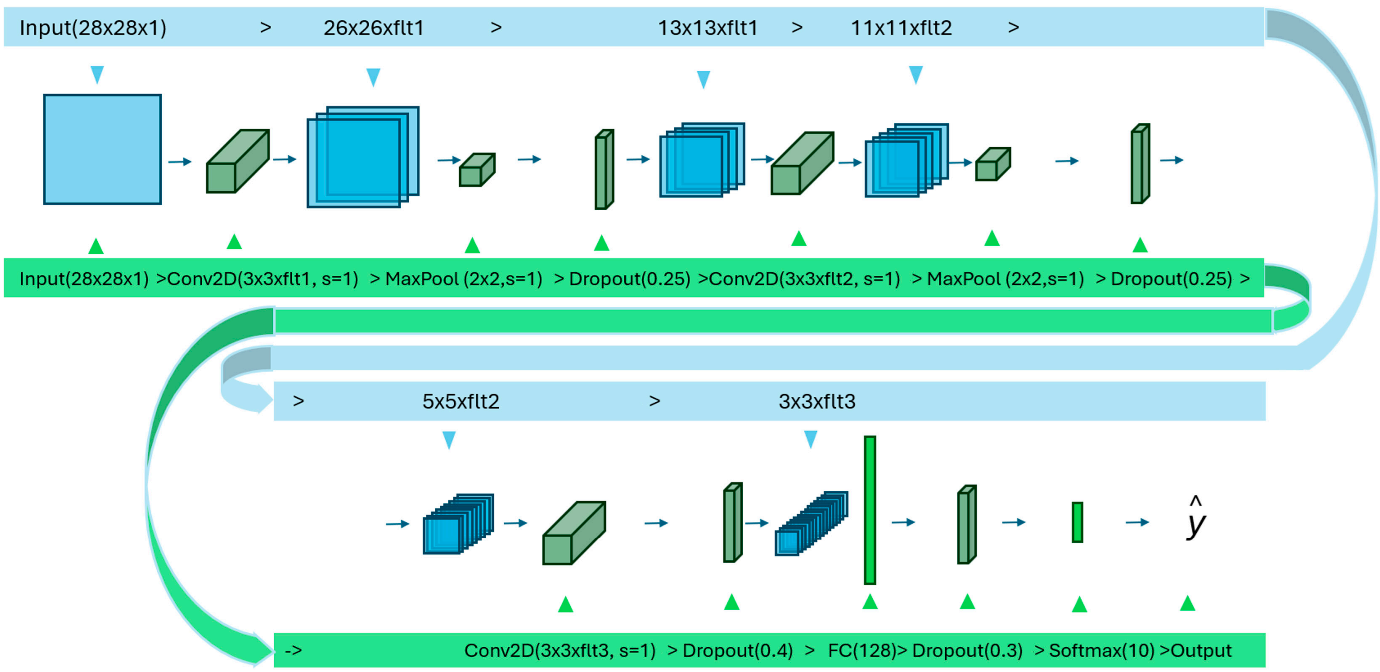

Figure 3. Architecture of CNN-3-128 convolutional network.

During the computational experiments, to evaluate the impact of model complexity on the maximum achievable quality, the number of convolution layer filters was successively changed from the maximum values of flt1 = 128, flt2 = 256, and flt3 = 512 to the minimum values of 2, 4, and 8. Accordingly, the number of adjustable model parameters was changed from 2,067,850 to 11,026. The Data Availability Statement contains a link to the full programme listing. Computational experiments were performed on a computer equipped with an Intel (R) Core (TM) i7-10TH processor with 64 GB of RAM and a discrete Nvidia Quatro T2000 graphics card.

5. Obtained Results

Table 2 summarises the results of the employment of some classical algorithms and ensembles of machine learning models. The results, which were improved compared to the literature data, are highlighted in bold. Most of the machine learning models were used with default parameters, and only some of them were slightly adjusted. The parameters for calling the models are given in the model column.

Table 2. Results of some classical and ensemble models and the results, obtained with the employment of the CNN-3-128 model.

Classifiers	**Accuracy**			**Model**
	1	**2**	**3**	
XGBoost [25]	85.3	**90**	88	Xgboost.XGBClassifier (nthread = 8)
DecisionTree [26]	79.8	79	78	DecisionTreeClassifier ()
ExtraTree [27]	77.5	**88**	87	ExtraTreesClassifier ()
GaussianNB [28]	51.1	**59**	51	GaussianNB ()
KNeighbors [29]	85.4	**86**	85	KNeighborsClassifier (n_neighbors = 5)
LogisticRegression	84.2	84	81	LogisticRegression ()
MLP [30]	87.1	**88**	**88**	MLPClassifier (random_state = 1, max_iter = 100)
RandomForest [31]	87.3	**88**	87	RandomForestClassifier (max_depth = 24, n_estimators = 200, random_state = 0)
SVC [32]	89.7	88	88	SVC ()
LightGBM [33]	-	**89**	88	Lgb.LGBMClassifier ()
HEM		89.56	88	HEM (MLP, XGBoost, LightGBM)
SEM		89.38	88	SEM (MLP, XGBoost, LightGBM)
CNN-3-128	-	**99.44**	**99.65**	Figure 3

N.B.

1. model accuracy is based on literature data
2. model accuracy is without image augmentation
3. model accuracy is using data augmentation.

Recently, Hybrid Ensemble Models (HEMs) and Stacking Ensemble Models (SEMs) have often been used in solving classification problems, which allow users one way or another to combine the results of several machine learning models. The results of using HEMs and SEMs are shown in Table 2. These models were built using MLP, XGBoost, and LightGBM. The CNN-3-128 model achieved a classification quality of 99.44 without applying augmentation with a total number of trained parameters 241,546 (flt1 = 32, flt2 = 64, flt3 = 128). The results of CNN-3-128 with different numbers of trained parameters at different numbers of training epochs, and with augmentation, are shown in Table 3.

Table 3. Training quality of models of different complexity.

	Trainable Params	Accuracy	Duration	Epochs
0	2,067,850	98.93000126	27.1199	2
1	2,067,850	99.29999709	74.6918	6
2	2,067,850	99.50000048	171.9928	14
3	2,067,850	99.50000048	367.5888	30
4	665,994	98.60000014	23.83484	2
5	665,994	99.29000139	69.86871	6
6	665,994	99.4599998	161.6529	14
7	665,994	**99.63999987**	344.0873	30
8	241,546	98.36000204	22.68729	2
9	241,546	99.12999868	67.69684	6
10	241,546	99.41999912	159.3603	14
11	241,546	99.57000017	363.4666	30
12	98,442	97.20000029	26.68739	2
13	98,442	98.62999916	79.62128	6
14	98,442	99.150002	176.73	14
15	98,442	99.30999875	355.7232	30
16	44,170	94.80000138	22.48629	2
17	44,170	97.46000171	67.65492	6
18	44,170	98.60000014	151.3534	14
19	44,170	98.87999892	317.5392	30
20	21,354	88.24999928	21.7845	2
21	21,354	94.34000254	64.72898	6
22	21,354	96.28000259	153.8319	14
23	21,354	97.13000059	323.0054	30
24	11,026	80.54999709	21.66025	2
25	11,026	86.82000041	63.86706	6
26	11,026	91.26999974	147.3278	14
27	11,026	93.16999912	309.7974	30

The demonstrated results were obtained with the following Image Data Generator parameters: rotation_range = 7.5, height_shift_range = 0.075, width_shift_range = 0.078, zoom_range = 0.085. When the number of training epochs was increased to 50, the model with 665,994 parameters showed the best result (accuracy = 99.65). At the same time, the model was wrong in 35 cases out of 10,000. The confusion matrix is shown in Figure 4.

The following 35 figures of the test set are not classified correctly: 359, 582, 659, 674, 882, 938, 1014, 1039, 1112, 1232, 1260, 1901, 2130, 2182, 2195, 2414, 2597, 3225, 3422, 3762, 3869, 3941, 3985, 4761, 4823, 5654, 5997, 6571, 6576, 6597, 6625, 8316, 8408, 9692, and 9729.

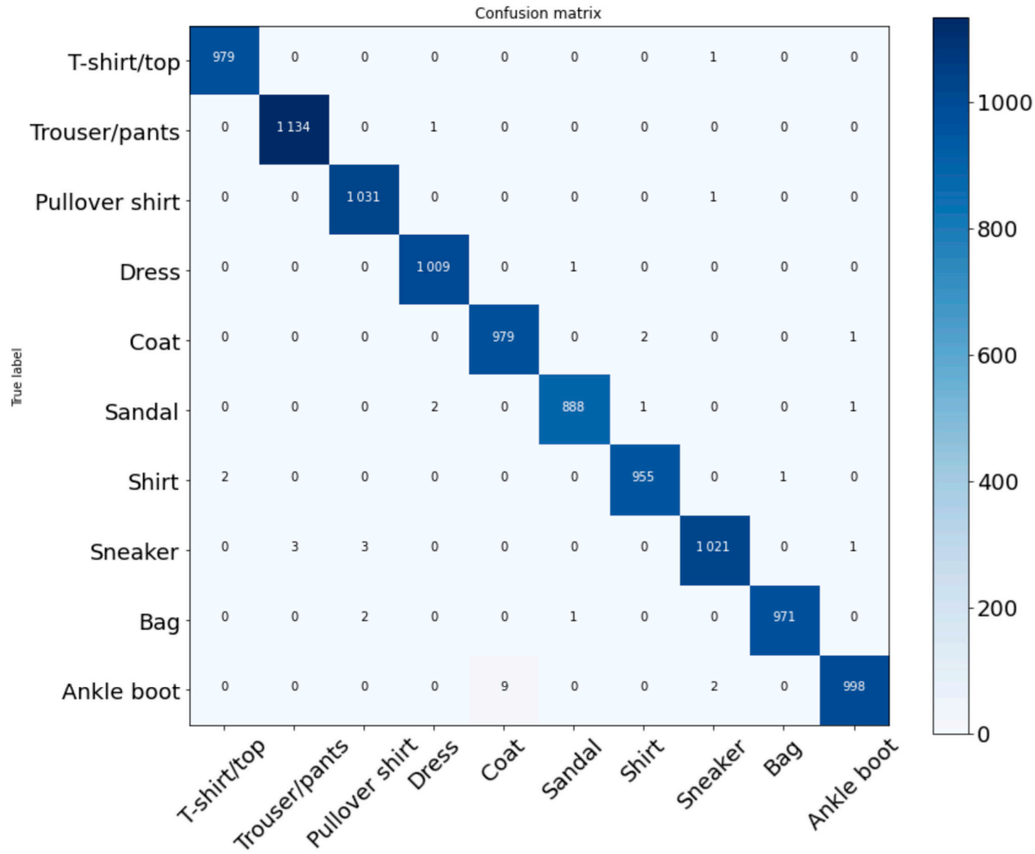

Figure 4. Confusion matrix obtained using CNN-3-128 (665,994) model (accuracy = 99.65).

6. Discussion of the Results

For most of the classical models, the obtained results are slightly better than those reported in the literature (see Table 2). It is expected that the data augmentation in the case of classical models does not cause an increase in the results, and even leads to a deterioration in classification quality due to the statistical nature of most classical models and decision tree ensembles.

Concerning the convolutional network model, there is reason to believe that the CNN-3-128 model has achieved a human-level quality of recognition, since it is extremely difficult to recognise the corresponding clothing items in misclassified images without prior training (see Figure 5). Although some images in Figure 5 look like numbers, they are actually clothes. The incorrectly classified items are mainly footwear (ankle boots, sneakers, sandals), in the amount of 22 pieces, as well as shirts (4), bags (3), and coats (3). analysing the confusion matrix, we can see that a large number of errors (9) are due to the model incorrectly classifying ankle boots as coats. Indeed, the corresponding images are similar. In addition, some styles have a very specific appearance. It can be assumed that higher-quality images are needed to recognise them.

In practise, on the one hand, this required a high learning, rate and on the other hand, it required a network architecture of minimum complexity in some resource-constrained applications of convolutional networks (for example, on board autonomous devices). It can be seen that models with more complex architectures reach high quality values faster (see Figure 6).

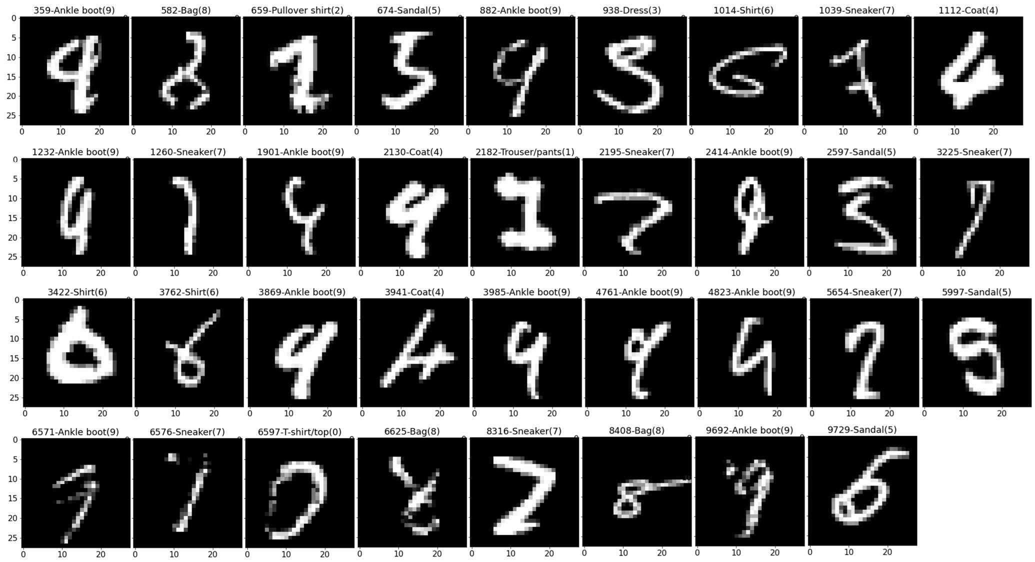

Figure 5. Misclassified clothing items.

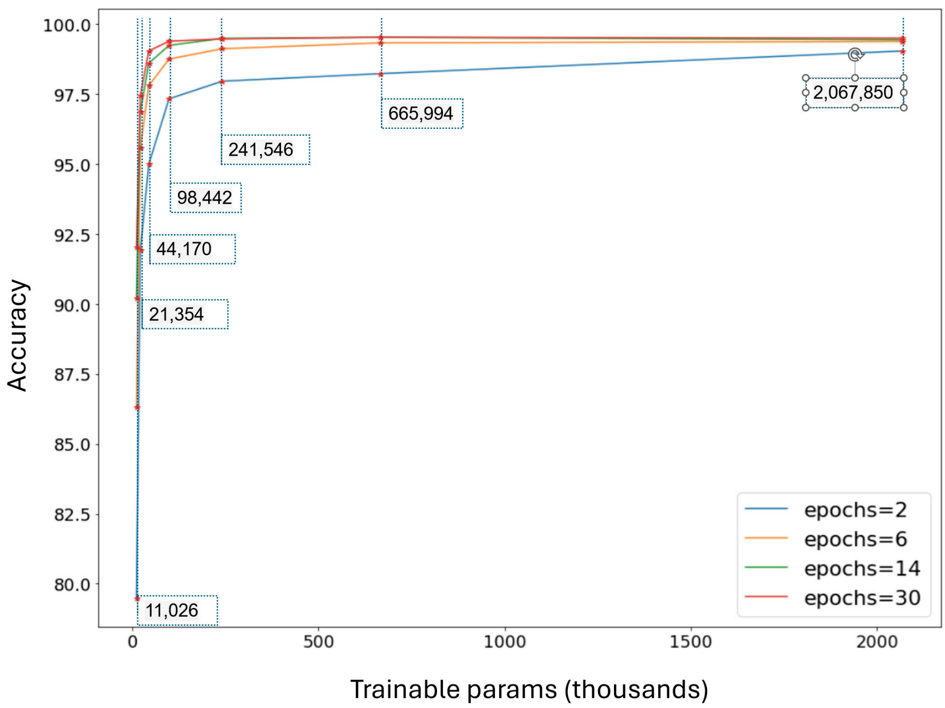

Figure 6. Quality of model classification with different complexities (from 11,026 to 2,067,850) with a different number of training epochs. The red dots show the results achieved by the models with the same number of epochs.

For example, the most complex model took two training epochs of 72 s to achieve the 99% accuracy value, while the simplified model with 44,170 trained parameters required 30 training epochs of 236 s to achieve the same result. It can be said that, in a similar way, a strong memory allows for a high learning rate for a given CPU-GPU-RAM configuration. This obviously allows us to recommend more complex architectures in situations where higher learning rates are required. Nevertheless, if memory resources are limited, it is possible to spend more time training a relatively simple model. On the other hand, it can be seen that simpler models with a number of parameters lower than a certain value (in this case, less than 98,000) are not able to achieve the ultimate classification quality. Note that the time spent on training models without using a GPU increases by about five times. Recently, transformer models have attracted much attention from researchers. These models have achieved great success in the field of natural language processing. They are characterised by a relatively simple structure, high scalability [34], and significant computational costs for training [35]. We have conducted experiments with GPT-4o-mini (zero-shot), considering it as a representative for transformer models. The results of the experiments are available at https://github.com/KindYAK/GPT-4o-vision-FashionMNIST-benchmark (accessed on 1 July 2024). The model showed a low result (accuracy 0.8). To improve the results, we used one of the simple visual transformer (ViT) models (see Data Availability Statement). The model, which we called ViT0fm, achieved an accuracy of 0.88. Obviously, the use of a ViT for such small datasets requires additional research.

7. Conclusions

The Fashion-MNIST image set is very popular. The best result achieved until recently in the classification of clothing items in this set was 99.1%, using the CNN-dropout-3 convolutional network [4]. In this paper, we propose the CNN-3-128 model, which achieves an accuracy of 99.44, and, with the use of image augmentation libraries, it outperforms this result and achieves the best classification result known to date for the Fashion-NMIST clothing image set. Although we used a relatively simple convolutional network model, image augmentation tuning required a rather large series of computational experiments, as a result of which we managed to select the optimal parameters of the augmenter. Using the data augmentation allowed us to achieve the classification result of 99.65% correctly classified clothing items. The achieved result shows that the use of a convolutional network with training dataset augmentation allows us to significantly improve the classification result. In turn, this allows us to expect that the proposed network architecture can be successfully applied in practical classification problems if the size of the input tensor is close to the size of the Fashion-MNIST images. The use of this architecture is not limited to clothing. In the following works, the author plans to use it to classify one-dimensional data obtained during well logging at uranium deposits described in [36]. Using the proposed convolutional network architecture, the relationship between the internal complexity of the model and the marginal quality was evaluated. The results show that the more complex the model, the faster it achieves high results. For example, larger models with more than 200 thousand trainable parameters achieve classification accuracy above 99% in 6 training epochs (approximately 70 s), while a simpler model (98,442 trainable parameters) requires 14 training epochs to achieve the same result (176 s). The training speed, of course, depends on the CPU-GPU-RAM configuration. It is expected that significantly simplified convolutional models with a number of trained parameters lower than 100,000 are not able to provide a marginal classification quality. The paper also presents updated results obtained with classical models. In most cases, we managed to slightly improve the results stated in the previous papers. However, as expected, the image augmentation of this set does not improve the result of classical machine learning models and different variants of decision tree ensembles. We would like to assume that the obtained results will be difficult to improve using CNN, especially considering the fact that misclassified clothing items cannot be identified manually either. In future experiments, it would be useful to compare the unrecognised images with those obtained with the CNN-dropout-3 model.

Funding: This research was funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan, under the following Grants: AP23488745 Rapid assessment of soil salinity using low-altitude unmanned aerial platforms (RASS); AP14869972 Development and adaptation of computer vision and machine learning methods for solving precision agriculture problems using unmanned aerial systems; BR21881908 Complex of urban ecological support (CUES); BR24992908 Support system for agricultural crop production optimization via remote monitoring and artificial intelligence methods (Agroscope); and BR18574144 Development of a Data Mining System for Monitoring Dams and Other Engineering Structures under the Conditions of Man-Made and Natural Impacts.

Data Availability Statement: The original contributions presented in the study are included in the article (https://www.dropbox.com/scl/fo/qtctnngb3pavezh4xvjdm/AKfKhOrcgahCfvdY0Z5zq2E?rlkey=ggpjy18lieo7wjdtudlzmv3jq&dl=0, accessed on 30 August 2024), further inquiries can be directed to the corresponding author.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A. Parameters of ImageDataGenerator

Table A1. ImageDataGenerator parameters used during computational experiments.

Parameters	Description
Rescale	The parameter allows for scaling the pixel values of the image. For example, rescale = 1/255 normalises the pixel values to a range of 0 to 1
Rotation_range	**The angle in degrees by which you can rotate the images randomly (Rotation_range = 0.75)**
Width_shift_range and height_shift_range	**The range of horizontal and vertical image shifts. Allows for creating random shifts of images (height_shift_range = 0.075, width_shift_range = 0.075)**
Brightness_range	The range of image brightness variation
Zoom_range	**Range of random image scaling (zoom_range = 0.085)**
Horizontal_flip и vertical_flip	Flips the image horizontally or vertically with a certain probability
Featurewise_center и samplewise_center	Normalisation of data by standard deviation of features or by individual samples
Zca_whitening	Application of ZCA whitening to reduce the correlation between pixels
Target_size	Size of the target images after transformations
Color_mode	Colour format of the input images (for example, "rgb" or "grayscale")
Batch_size	Number of images processed per iteration

N.B. Parameters used to achieve the best classification results are highlighted bold.

References

1. LeCun, Y. The MNIST Database of Handwritten Digits. 1998. Available online: http://yann.lecun.com/exdb/mnist/ (accessed on 2 August 2024).
2. Yadav, C.; Bottou, L. Cold Case: The Lost MNIST Digits. Advances in Neural Information Processing Systems. Curran Associates, Inc. *arXiv* **2019**, arXiv:1905.10498.
3. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv* **2017**, arXiv:1708.07747.
4. Leithardt, V. Classifying garments from fashion-MNIST dataset through CNNs. Advances in Science. *Technol. Eng. Syst. J.* **2021**, *6*, 989–994.
5. Shen, S. Image classification of Fashion-MNIST dataset using long short-term memory networks. *Res. Sch. Comput. Sci.* **2018**. Available online: https://users.cecs.anu.edu.au/~Tom.Gedeon/conf/ABCs2018/paper/ABCs2018_paper_38.pdf (accessed on 30 August 2024).
6. Samia, B.; Soraya, Z.; Malika, M. Fashion images classification using machine learning, deep learning and transfer learning models. In Proceedings of the 2022 7th International Conference on Image and Signal Processing and their Applications (ISPA), Mostaganem, Algeria, 8–9 May 2022; IEEE: New York, NY, USA; pp. 1–5.
7. Nocentini, O.; Kim, J.; Bashir, M.; Cavallo, F. Image classification using multiple convolutional neural networks on the fashion-MNIST dataset. *Sensors* **2022**, *22*, 9544. [CrossRef] [PubMed]
8. Rohrmanstorfer, S.; Komarov, M.; Mödritscher, F. Image Classification for the Automatic Feature Extraction in Human Worn Fashion Data. *Mathematics* **2021**, *9*, 624. [CrossRef]
9. Kiapour, M.; Yamaguchi, K.; Berg, A.; Berg, T. Hipster wars: Discovering elements of fashion styles. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, I. pp. 472–488.
10. Chen, J.C.; Liu, C.F. Deep net architectures for visual-based clothing image recognition on large database. *Soft Comput.* **2017**, *21*, 2923–2939. [CrossRef]

11. Bossard, L.; Dantone, M.; Leistner, C.; Wengert, C.; Quack, T.; Van Gool, L. Apparel classification with style. In Proceedings of the Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Republic of Korea, 5–9 November 2012; Revised Selected Papers, IV. pp. 321–335.
12. Donati, L.; Iotti, E.; Mordonini, G.; Prati, A. Fashion Product Classification through Deep Learning and Computer Vision. *Appl. Sci.* **2019**, *9*, 1385. [CrossRef]
13. Liu, Z.; Luo, P.; Qiu, S.; Wang, X.; Tang, X. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1096–1104.
14. An, H.; Lee, K.; Choi, Y.; Park, M. Conceptual framework of hybrid style in fashion image datasets for machine learning. *Fash. Text.* **2023**, *10*, 18. [CrossRef]
15. Becker, K. Image Recognition for Fashion with Machine Learning. 2017. Available online: https://www.primaryobjects.com/kory-becker/ (accessed on 2 August 2024).
16. Shubathra, S.; Kalaivaani, P.; Santhoshkumar, S. Clothing image recognition based on multiple features using deep neural networks. In Proceedings of the 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2–4 July 2020; IEEE: New York, NY, USA; pp. 166–172.
17. Bhatnagar, S.; Ghosal, D.; Kolekar, M. Classification of fashion article images using convolutional neural networks. In Proceedings of the 2017 Fourth International Conference on Image Information Processing (ICIIP), Shimla, India, 21–23 December 2017; IEEE: New York, NY, USA; pp. 1–6. [CrossRef]
18. Shin, S.Y.; Jo, G.; Wang, G. A novel method for fashion clothing image classification based on deep learning. *J. Inf. Commun. Technol.* **2023**, *22*, 127–148.
19. Meshkini, K.; Platos, J.; Ghassemain, H. An Analysis of Convolutional Neural Network for Fashion Images Classification (Fashion-MNIST). In *Proceedings of the Fourth International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'19). IITI 2019*; Advances in Intelligent Systems and Computing; Kovalev, S., Tarassov, V., Snasel, V., Sukhanov, A., Eds.; Springer: Cham, Switzerland, 2020; Volume 1156, pp. 85–95. [CrossRef]
20. Nguyen, M.; Nguyen, H. Clothing Classification Using Shallow Convolutional Neural Networks. In *Biomedical and Other Applications of Soft Computing*; Springer: Cham, Switzerland, 2022; pp. 239–250.
21. Greeshma, K.V.; Sreekumar, K. Hyperparameter Optimization and Regularization on Fashion-MNIST. *Classif. Int. J. Recent T Echnology Eng.* **2019**, *8*, 3713–3719.
22. Seo, Y.; Shin, K. Hierarchical convolutional neural networks for fashion image classification. *Expert Syst. Appl.* **2019**, *116*, 328–339. [CrossRef]
23. Vijayaraj, A.; Vasanth Raj, P.; Jebakumar, R.; Gururama Senthilvel, P.; Kumar, N.; Suresh Kumar, R.; Dhanagopal, R. Deep learning image classification for fashion design. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 7549397. [CrossRef]
24. Kayed, M.; Anter, A.; Mohamed, H. Classification of Garments from Fashion MNIST Dataset Using CNN LeNet-5 Architecture. In Proceedings of the 2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE 2020), Aswan, Egypt, 8–9 February 2020; pp. 238–243.
25. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
26. Quinlan, J. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
27. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3–42. [CrossRef]
28. Bayes, T. An essay towards solving a problem in the doctrine of chances. *Biometrika* **1958**, *45*, 296–315. [CrossRef]
29. Fix, E. *Discrimination Analysis: Nonparametric Discrimination, Consistency Properties*; US Air Force School of Aviation Medicine, University of Iowa: Iowa City, IA, USA, 1985; Volume 1, p. 42.
30. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [CrossRef]
31. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
32. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
33. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3149–3157.
34. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in vision: A survey. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 1–41. [CrossRef]
35. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
36. Mukhamediev, R.; Kuchin, Y.; Yunicheva, N.; Kalpeyeva, Z.; Muhamedijeva, E.; Gopejenko, V.; Rystygulov, P. Classification of Logging Data Using Machine Learning Algorithms. *Appl. Sci.* **2024**, *14*, 7779. [CrossRef]