VIT

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

FALL SEMESTER 2021-22

SOFT COMPUTING (ITE1011)

Final Review

Kunj Patel	19BIT0256	*Under the guidance of*
Robin Singh	19BIT265	Dr. Balakrushna Tripathy
Sejal Bharti	19BIT286	Higher Academic Grade, SITE

Email ID: kunj.patel2019@vitstudent.ac.in

Classification of Fashion Apparels Using Convolutional Neural Network Algorithm on MNIST Dataset

ABSTRACT

The primary focus of this research is to investigate the accuracy of an artificial neural network(ANN) approach for estimating the energy dissipation of the skimming flow regimeover stepped spillway because of the imprecise, insufficient, ambiguous and uncertain dataavailable.

Convolutional neural network (CNN) has applied to estimate the energy dissipation since they are capable of correlating large and complex data sets without any prior knowledge ofthe relationships among them. CNN (Convolutional neural network) is applied to classify thedataset. The 10 classes are as follows: 0 => T-shirt/top 1 => Trouser 2 => Pullover 3 => Dress4 => Coat 5 => Sandal 6 => Shirt 7 => Sneaker 8 => Bag 9 => Ankle boot

INTRODUCTION

The garment industry houses an enormous amount and sort of data. At every checkpoint ofthe supply chain, information is collected and stored by each actor. This data, when used intelligently, can help with solving an honest deal of problems for the industry.

As the contemporary customer relies on online retail channels to form purchases, the necessity also arises for powerful and intelligent systems which will recommend, personalize,or help the customer in making purchasing decisions. Such models (decision support systems)can help customers find the proper garments, consistent with their requirements. The first step towards achieving this is often to form the models recognize the various garment categories and corresponding garment attributes.

Technically, deep learning CNN models to coach and test, each input image will pass it through a series of convolution layers with filters (Kernels), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and
1. Computers sees an input image as array of pixels and it depends on the image resolution.Based on the image resolution, it'll see h x w x d(h = Height, w = Width, d = Dimension).

Language and Platform

For this project we have used google collab and Python3 language.

LITERATURE REVIEW

1.Convolutional Neural Networks for Fashion Classification and Object Detection

ABSTRACT: The identification of clothing items in a photograph is referred to as fashion categorization. Social networking, e-commerce, and criminal law all have applications in this subject. In our line of employment,

Within the fashion classification umbrella, we concentrate on four tasks:

(1) grouping of clothing types into multiple classes.
(2) a classification of garment attributes; and a retrieval of apparel.
(3) Detection of clothing objects; and
(4) detection of nearest neighbors.
We report garment style categorization accuracy measures (50.2 percent) and the categorization of clothing attributes (74.5 percent) can surpass literature baselines for the linked datasets. We also report on some encouraging qualitative results. our clothes retrieval and clothing object detection results tasks.
The task of locating the most similar clothing items to a query clothing item is known as clothing retrieval. When combining characteristics learnt from both (a) the clothing type dataset and (b) the clothing attribute dataset, we expect that the nearest neighbors will more closely resemble the query image. Although we expect some overlap in the features learnt for the two datasets,
we also expect each dataset to have its own set of unique features and weightings, thus integrating the data will result in a more robust collection of weights.
Clothing Object Detection is the process of identifying specific regions of clothing objects in an image.
Clothing item detection, for example, entails predicting bounding boxes that would capture the unique articles of clothing such as the shirt, pants, and shoes given an image of a human wearing a full outfit.
Using a Nearest Neighbors method on the raw pixels in the image is one way to solve this problem. This is a highly limited method because it compares images pixel by pixel to locate those that match on a per pixel basis. Instead, we employed context-based characteristics to describe each image by looking at the activations from the 7th layer of the fine-tuned CaffeNet convolutional network.
Each feature in this option indicates a high-level element of the input pixel features that was not captured in the original technique by a single pixel.

2. A Survey of Image Classification Methods and Techniques

I. ABSTRACT: This paper examines the current state of image categorization strategies and techniques. Image classification is a complicated process that is influenced by a number of factors. The current methodologies, difficulties, and future possibilities of picture classification are discussed here. Advanced classification techniques will be the main focus, as they are employed to improve classification accuracy. In addition, certain critical aspects concerning categorization performance are discussed.

II. DIFFERENT TYPES OF IMAGE CLASSIFICATION TECHNIQUES

A. Based on the information acquired from different sensors: On medium resolution multi-temporal images, the majority of the classifiers perform admirably (like Landsat TM). It is in charge of correctly classifying vegetation types based on variations in plant features. When we combine the SAR and optical pictures, however, classification becomes more difficult because there are more than 200 bands to consider. Advanced approaches that can handle difficult challenges can be used to gain accurate categorization of hyper spectral images. Because of the modest ratio between the size of the input feature space and the number of training samples, the problem arises.

As a result of this issue, we have inaccurate estimations of classifier parameters, which leads to low labelling accuracy and erroneous generalization properties. As a result, new systems that can take high-resolution photos from satellites and airborne platforms have been developed. These devices are capable of acquiring: (a) high-resolution multispectral images with geometric resolutions on the order of (or less than) 1m; and (b) hyper-spectral images with hundreds of bands associated with narrow spectral channels.

B. Based on nature of training sample used in classification: Supervised classification: Prior information is required before testing in supervised classification, and it must be obtained by the analyst. The supervised classification technique involves the following steps: i. Identifying the training regions for each informative class.

ii. Signatures are used to identify people (variance, covariance, mean etc)

iii. After that, all pixels are categorized.

iv. The informational class is mapped out.

The fundamental benefit of supervised categorization is that it allows an operator to notice and remedy errors. This method has the disadvantages of being time consuming and expensive.

Furthermore, the analyst's training data may not emphasize all of the conditions shown throughout the image, making it vulnerable to human mistake.

C. Based upon the nature of spatial information: Spectral classifiers: For picture classification, pure spectral information is employed. Due to a large degree of variation in the spatial distribution of the same class, a 'noisy' categorization result. Minimum distance, maximum likelihood, and artificial neural network are some examples.

Contextual classifiers: In this example, image categorization is based on spatially nearby pixel information. Contextual correction from point to point, iterated conditional modes, and a frequency-based contextual classifier are some examples.

3. Classification of Fashion Article Images using Convolutional Neural Networks:

ABSTRACT: We offer a state-of-the-art model for classification of fashion article photos in this research. To categorize photos in the Fashion-MNIST dataset, we built convolutional neural network-based deep learning architectures. For ease and speed of learning, we suggested three distinct convolutional neural network designs and used batch normalization and residual skip connections. On the Fashion-MNIST benchmark dataset, our model performs admirably.

Comparing our suggested model to existing state-of-the-art systems in the literature, we find that it improves accuracy by about 2%.

I. PROBLEM DEFINITION: Picture classification is a fundamental topic in computer vision that has a wide range of applications, including image and video indexing. Although identifying a visual entity from an image is a relatively simple challenge for a human to solve, it is extremely difficult for a computer system to do so with human-level accuracy. In order to correctly recognise and classify the photos, the method must be invariant to a lot of modifications. Different lighting conditions, scale and viewpoint alterations, deformations, and occlusions, for example, may cause the system to incorrectly anticipate the image class.

II. PROPOSED METHODOLOGY: Convolutional neural networks are based on neurobiology.

A convolutional network's typical layer consists of three stages. In the first stage, we construct a feature map by sliding a large number (tens to thousands) of filters or kernels over the input image, which are typically 3x3, 4x4, or 5x5 in size. We add up the element wise dot product of the filter values and the section of the picture we're sliding over as we slide the kernel over it.

It is a very memory efficient technique because the same kernel is used across the image. Because the kernels in a layer are independent of one another, results in a graphics processing unit can be computed exceptionally quickly (GPU).

The convolution operation between a two-dimensional image I and a two-dimensional kernel K is,

III. VISUALIZATION: Various approaches for visualization and understanding of convolutional nets have been developed in literature. One of the most popular strategy is to visualize the convolution layer outputs of an input image during forward propagation.
This method will generally show different boundaries, corners in lower-level convolutional layers, and gradually more higher-level features in the deeper convolutional layers. Another very common strategy is to visualize the weights of the network.
This is useful because well-trained networks usually display smooth and nice kernels without any noisy patterns. Noisy structures can indicate that the network most probably has not been trained for enough time, or possibly a low regularization factor which has led to overfitting.

IV. CONCLUSION: Convolutional neural networks are based on neurobiology.
A convolutional network's typical layer consists of three stages. In the first stage, we construct a feature map by sliding a large number (tens to thousands) of filters or kernels over the input image, which are typically 3x3, 4x4, or 5x5 in size. We add up the element wise dot product of the filter values and the section of the picture we're sliding over as we slide the kernel over it.
It is a very memory efficient technique because the same kernel is used across the image. Because the kernels in a layer are independent of one another, results in a graphics processing unit can be computed exceptionally quickly (GPU).

4. Classification of Garments from Fashion MNIST Dataset Using CNN LeNet-5 Architecture:

I. ABSTRACT: Deep learning has been used to a wide number of fields. Convolutional neural networks are a type of deep neural network that produces the most consistent results when tackling real-world situations. Network of Neurons (CNN). CNN has been used by the fashion industry on several occasions. utilizing e-commerce to handle a variety of issues, such as clothing Recognize, search for garments, and make a recommendation. A crucial phase in the process Image categorization is used in all these solutions. However, Clothing classification is a difficult process because there are so many different types of clothing. The complexity of clothing categorization is important. complex. Because of the complexity of the depth, multiple classes are created. have a lot of similarities, which makes it difficult to classify them. It becomes quite difficult. The CNN-based LeNet-5 is used in this paper. It is proposed that architecture be used to train CNN parameters on. MNIST dataset on fashion. The findings of the experiments demonstrate that LeNet-5 is effective. The accuracy of the model was over 98 percent. As a result, it outperforms. both the old CNN model and various current state-of-the-art models Literatures' models.

II. MATERIAL AND METHODS: Convolutional neural network is a class of deep feed-forward artificial neural network which is used mainly for image processing, classification, segmentation, and others. It includes three types of layers: convolutional, pooling and fully connected layers.
First Layer (C1): The input for this layer is a 32×32 gray scale image. This image passes through the first convolutional layer with six feature maps or filters having size 5×5 and a stride of 1. The image dimensions will be changed from 32x32x1 to 28x28x6.
Second Layer (S2): It is an average pooling layer with six feature maps of size 14×14. Each unit in each feature map is linked to a 2×2 block in the identical feature map in C1. Also, S2 has 12 trainable parameters and 5880 connections. The resulting image dimensions will be reduced to 14x14x6.
Third Layer (C3): Convolutional layers with 16 feature maps having size 5×5 and a stride of 1. Each unit in each feature map is linked to different 5×5 blocks as like positions in a subset of S2’s feature maps. Finally, the last one loads input from all S2 feature maps.

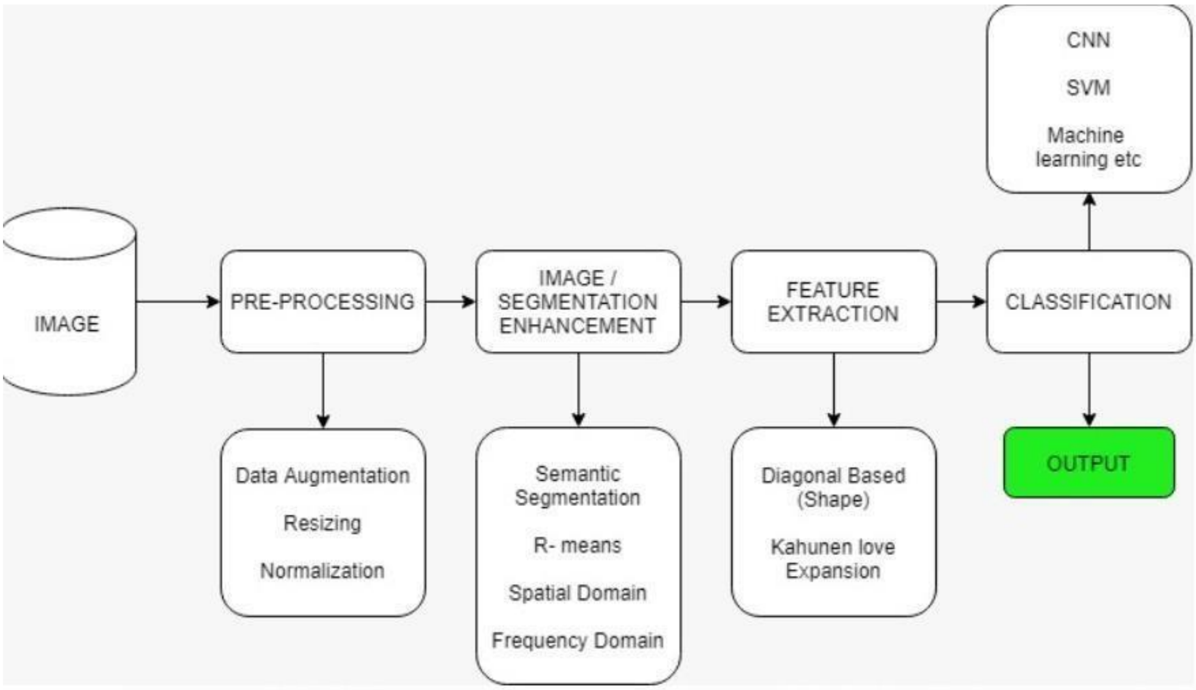
Fourth Layer (S4): Also, it is an average pooling layer with filter size 2×2 and a stride of 2 with 16 feature maps of size 5×5. Each unit in each feature map is linked to a 2×2 blocks in the identical feature map in C3, and identical path as C1 and S2. This layer is the same as the second layer (S2) unless it has 32 trainable parameters and 2000 links to the output will be decreased to 5x5x16.
Fifth Layer (C5): Fully connected convolutional layers with 120 feature maps. Each unit is linked to a 5× 5 block on all 16 of S4's feature maps. C5 is classified as a convolutional layer instead of a fully connected layer because if LeNet-5 input were made bigger with everything else protected fixed, the feature map dimension would be larger than 1×1. Each of the 120 units in C5 is linked to all the 400 nodes (5x5x16) in the fourth layer S4.
Sixth Layer (F6): A fully connected layer with 84 units. It is fully linked to C5. It contains 10164 trainable parameters.
Output Layer: A fully connected SoftMax layer with 10 possible rates identically to the digits from 0 to 9.

III. CONCLUSION:

With the advancement of deep learning approaches, image identification using CNN has become increasingly popular in the fashion industry. such as clothing classification, retrieval, and automatic retrieval Labeling of clothing We use the LeNet-5 architecture in this paper. The Fashion MNIST dataset is a collection of data about fashion. The performance of LeNet-5 is improved. models that already exist We intend to carry out a thorough comparison. on the basis of alternative CNN architectures (such as VGG16) datasets on clothing (such as Image Net).

BACKGROUND

GENERAL ARCHITECTURE



PRE-PROCESSING TECHNIQUES

Binarization:

Binarization is that the process of remodeling data features of any entity into vectors of binary numbers to form classifier algorithms more efficient. during a simple example, transforming an image's gray-scale from the 0-255 spectrum to a 0-1 spectrum is binarization.

Thresholding:

The simplest thresholding methods replace each pixel in a picture with a black pixel if the image intensity is a smaller amount than some fixed constant T, or a white pixel if the image intensity is bigger than that constant. within the example image on the proper, this leads to the dark tree becoming completely black, and therefore the white snow becoming completely white.

Normalization:

Linear Normalization is that the process that changes the range of pixel values.

The benefits of normalization include:

Searching, sorting, and creating indexes is quicker , since tablesare narrower, and more rowsfit on a knowledge page.

Data Augmentation

Data Augmentation approaches overfitting from the basis of the matter , the training dataset. are often "> this is often done under the idea that more information can be extracted from the first dataset through augmentations

Image Segmentation

K-means Clustering

Using global K-means for determining clothing segments worked relatively well given its simplicity. For our dataset, this type of study yielded 67% accuracy. Furthermore, when including a neighborhood detection algorithm, this method proved to be useful in providing reasonable results. However, as implemented, this method didn't provide sufficient consistency, generalizability or precision.[1]

Segmentation through Combinatorial Computer Vision and Machine Learning Algorithms

Using our learning algorithm we were ready to successfully resolve clothing from nearly half the pictures in our data set. However, once we adjusted the thresholds and bin sizes we were ready to resolve clothing from all of the pictures in our training sample. Accordingly, we believe that by modifying our learning algorithm to automatically determine the simplest parameters, this process are often extended to incorporate very large training sets with average accuracy of 76%.[1]

Supervised Semantic Segmentation

MIM to jointly estimate the parameters of local appearance models for the semantic classes and to infer latent superpixel labels. For segmenting test images, we integrate them into MIM by means of a learned multiple kernel image similarity.[2]

STF based ψ, ϕ on S and $\mu+$		test		train	
MIM	objectness	average	total	average	total
-	-	53	46	51	63
yes	-	55	58	66	70
-	yes	59	56	77	70
yes	yes	**67**	67	83	80

We presented a weakly supervised semantic segmentation method that, for the first time, can compete with fully supervised ones.[2]

Image Enhancement

Spatial domain technique

The advantage of spatial domain technique is that it's simple to understand and thus the complexity of these techniques is extremely low which helps in real time implementation.[3]

The disadvantages of spatial domain technique are that it doesn't provides adequate robustness and perceivably.[3]

Frequency domain technique

Whereas Frequency domain technique having advantages which include low computation complexity, easy to seem at, manipulation of images frequency composition and thus the special transformed domain property is certainly applicable.[3]

disadvantage of Frequency Domain is that it cannot enhance properly every a neighborhood of an image simultaneously and thus the automation of image enhancement is additionally very difficult.[3]

Image illumination

The purpose of LightenNet is to seek out out a map132 ping, which takes a weakly illuminated image as input and out133 puts its illumination map that's subsequently used to obtain 134 the improved image supported Retinex model.[4]

Fig. 4. Qualitative comparisons on synthetic images. (a) Synthetic weakly illuminated images. (b) Results of HE (Gonzalez and Woods, 2017). (c) Results of MSR (Jobson et al., 1997). (d) Results of AWVM (Fu et al., 2016b). (e) Results of LIME (Guo et al., 2017). (f) Results of our method. (g) Ground truth.

Fig[4]

Validation Techniques

1. Train/test split

The most basic method is that the train/test split. The principle is straightforward , you merely split your data randomly into roughly 70% used for training the model and 30% for testing the model.

The advantage of this approach is that we will see how the model reacts to previously unseen data. Proposed Algorithm.

2. k-fold cross-validation

This is where k-fold cross-validation comes in. It splits the info into k folds, then trains the info on k-1 folds and test on the one fold that was overlooked . It does this for all combinations and averages the result on each instance.
The advantage is that each one observation is used for both training and validation, and every observation is employed once for validation. We typically choose either i=5 or k=10 as they find a pleasant balance between computational complexity and validation accuracy.

3. Nested Cross-Validation

When you are optimizing the hyperparameters of your model and you employ an equivalent k-Fold CV strategy to tune the model and evaluate performance you run the danger of overfitting. you are doing not want to estimate the accuracy of your model on an equivalent split that you simply found the simplest hyperparameters for.

Classification

ANN

Authors and Year	Methodoligy	Advantages	Issues	Metrics
Barış Gülbaş1 *Abdulkadir Şengür* *Emre* *İncel1* *2019[1]*	Convolution Neural Network (CNN) Effective Learning machines (ELM)	used due to its robustness and simple structure. obtained results show that the deep learning improves the fashion classification.	significantly slower due to an operation such as maxpool. A ConvNet requires a large Dataset to process and train the neural network	Accuraccy Precision
Guanglei Zhang *Yongsheng Ding* *2018* *[2]*	sparsity auto-encoder (SAE) CNN	More precise images are that they are more location invariant	CNN has several layers then the trainingprocess takes a lot of time if the computer doesn't consist of a good GPU.	Accuraccy Precision

Kyung-shik Shin *Yian Seo* *2019* *[3]*	CNN	The pre-training and fine-training enable us to supplement the small quantity of dataset and reduce the training hours into a few minutes without using GPU	Its hard to collect large enough dataset hard to attain a quality image dataset	Train accuracy Final Test accuracy Validation Accuracy Loss
Rasmus Rothe Marko Ristin Matthias Dantone Luc Van Gool 2018[4]	HOG hierarchical k-means clustering. supervised	using in supervised manner increases acccuracy by 16% does not require powerful dedicated hardware.	only for low and mid level features Less successful results and accuraccy than CNN	Accuracy F-measure
Alexander Schindler Thomas Lidy 2018[5]	CNN InceptionV3: vgg 16 and vgg 19	high accuracy large data set	cannot differentiate minor features orsub catagories such as jeans and skinny jeans	Accuracy Confusion Matrix
Geoffrey E. Hinton **Ilya Sutskever** **Alex Krizhevsky** 2019[6]	modified CNN dropout	To make training faster, we used nonsaturating neurons and a very efficient GPU implementation	error rate is not negligible Apparels can easily deform	**Accuracy** **Error rate**

		of the convolution operation. To reduce overfitting in the fully connected layers we employed a recently developed regularization method called “dropout” that proved to be very effective.	due to their material	
John Paul A. Madulid Paula E. Mayol 2019[7]	CNN	easy to implement simple architecture	hard to attain a quality image dataset	Accuracy
Jianbing shen, **Wenguan** **wang2018[8]**	Bidirectional Convolutional Recurrent Neural Network (BCRNN)	Efficient Architecture	Not enough/adequate dataset.	**Accuracy andprecision**
Shobit Bhatnagar **,** **Deepanway Ghosal** **2017[9]**	CNN BatchNorm	Large dataset And large variety		**Confidenc escore** **accuracy**
Majuran Shajini and Amirthalingam Ramanan (2020) **[10]**	CNN	attention pipelines allow the model to represent multiscale contextual information of landmarks, thus	Time consuming	**Efficiency** **,accuracy**

		improving the efficiency of classification.		
Brian Lao, Karthik Jagadeesh **(2016)[11]**	CNN	Simplified architecture Good accuracy	Not the best global solution	**accuracy Precision F1 score**

Jan Cychnerski Adam Brzeski 2017	Convolution Neural Network , ELM	Identifies diff types of apparels	Need large data-set to process and train neural network.	**F score accuracy**
Noel J kadouch (2019) [12]	H-CNN, Deep learning	Faster identification, Good performance	Simplification is Concerned, Highcomplexity	**Accuracy**
Alexander Schindler, Stephan Karner **(2018)[13]**	Deep learning, CNNs	85.6% accuracy indetermining the classification.	Small Data-set	**Accuracy But small dataset**
Dr. Vaibhav kumar (2020) [14]	Convolutional Neural Network (CNN) model	More efficient and reliable 98% in recognizin gtrousers, more than 85% accuracy in recognizing apparels.		**Accuracy Confusio nmetrix**
Rohit Patki, Suhas Suresha **(2017) [15]**	SVM, Convolutional Neural network (CNNs)	Trained on different models, Has potential of doing better with higher resolution images	Not enough/adequate dataset.	**Accuracy**

Authors &Year	Methodology or Techniques used	Advantages	Issues	Metrics used
Rohit Patki, SuhasSuresha (2017)	SVM, Convolutional Neural network (CNNs)	Trained on different models, Has potential of doing better with higher resolution images	Not enough/adequate dataset.	**Accuracy**
Seo, yian Shin,kyung shik 2018	CNN, machine learning	Large-scale dataset Easy to implement	Accuracy of lessthan 76%	**Recall, AccuracyError rate**
Rohit Patki, SuhasSuresha (2019)	SVM, CNN	Higher accuracy	Style and Attribute Classification	Confusion metrix
YepingPeng, JunhaoCai, TonghaiWu, Guangzhong Cao 2019	CNN, Transfer learning, SVM	Identification of diff. types of apparel		Support vector, accuracy, precision
Rasmus Rothe, Marko Ristin, Matthias Dantone, Luc Van Gool (2018)	OG, hierarchical k-means clustering., supervised	using in supervised manner increases accuracy by 16%, does not require powerful dedicated hardware.	only for low and mid-level features, less successful results and accuracy than CNN	Accuracy F-measure
Lukas Bossard1, Matthias Dantone, Till Quack (2019)	Support Vector Machines (SVMs), Random Forest	easily handle high dimensional data , inherently multi-class classifiers	SVM does not perform very well when the data set has more noise, Classification are difficult to interpret	Avg Accuracy, Precision

Machine Learning

Seo, yian Shin, kyung shik 2018	**machine learning**	**Large-scale dataset Easy to implement**	**Accuracy of less than 76%**	**Recall, Accuracy Error rate**
Liu, L.; Su, J.; Zhao, B.; Wang, Q.;	Decision trees	Can be used both for classification and regression	Leads to overfitting	precision
Sheenam Jain and Vijay Kumar	**Naïve Bayes**	Fast efficient and easy to implement	Works on assumption of independence of features	Precision, accuracy
Mao, W.; Wang, F.-Y. Cultural Modeling for Behavior	Random forest	Runs efficient on large dataset Lower risk of overfitting	High Variance (Model isgoing to change quickly with a change in training data)	precision

ANN

- **LSTM**

  • I test the normal RNN model and therefore the LSTM model respectively. The result shows that LSTM model performs better than the RNN model with a betteraccuracy at 80.20%, while the RNN model can only achieve a 74.52% accuracy [17]

  • To overcome the weakness of traditional RNN, i exploit the Long-Short Term Memory (LSTM) technique to create the model. I optimize the model by fine tuning,cross validation, Network Pruning and Heuristic Pattern Reduction method. Finally, the accuracy of LSTM model can reach 89.94% with acceptable time consumption
- CNN

  • AlexNet,VGGNet,ResNet

  ELM doesn't need any optimization step, which makes it strong against the quality back- propagation learning algorithm. The obtained results show that ResNet features are moreefficient than AlexNet and VGGNet features

- GoogLeNet

pre-train the GoogLeNet architecture on ImageNet dataset and fine-tune on our fine-grained fashion dataset supported design attributes. this may complement the tiny size of dataset and reduce the training time.[22]

The overall train accuracy points 0.79 and the overall validation accuracy points 0.62. The final test accuracy rates 0.62 and the average cross-entropy loss is 1.03.[22]

CNN-RNN

It can exploit the relationship between the coarse and fine categories, which, in turn, helps the traditional image classification task. For example, when we build the CNN-RNN framework with wrn-28-10 [53], we can increase the accuracy of coarse and fine categories by 2.8% and 1.68%, respectively.

CNN with k-means

The results are summarized . the typical classification accuracy supported hand-crafted HOG and SIFT features achieve average accuracies of fifty .1% and 62.0%.

Table 1: Our method outperforms the hand-engineered features, e.g. 10% higher accuracy compared to SIFT.

	bags	underwear	coats	jewellery	blouses	watches	hats	belts	tops	pants	skirts	jumpers	jumpsuit	dresses	glasses	**avg.**
HOG	57.6	20.4	42.4	35.8	48.4	92.0	69.6	70.8	5.6	81.8	32.6	33.0	46.6	18.8	95.6	**50.1**
SIFT	83.2	39.4	55.4	46.2	35.0	91.4	70.2	79.4	21.2	89.8	58.0	50.6	59.0	56.8	94.8	**62.0**
Ours, unsupervised	78.4	50.6	67.0	61.4	37.8	94.4	77.8	83.6	49.0	90.2	72.4	50.4	72.2	73.0	92.6	**70.1**
Ours, supervised	81.8	53.8	68.8	63.8	42.8	95.6	81.2	88.4	48.8	90.8	74.2	49.6	73.4	72.2	94.2	**72.0**

While the authors of [6] train low- and mid-level features with k-means in an unsupervised manner, we show how k-means clustering are often supervised to realize better accuracy.

The accuracy relatively increases by 16.1% when using our low-level features rather than the pre-designed ones.

The results are summarized. The average classification accuracy based on hand-crafted HOG and SIFT features achieve average accuracies of 50.1% and 62.0%.

While the authors of [6] train low- and mid-level features with k-means in an unsupervised manner, we show how k-means clustering can be supervised to achieve better accuracy.

The accuracy relatively increases by 16.1% when using our low-level features instead of the pre-designed ones.

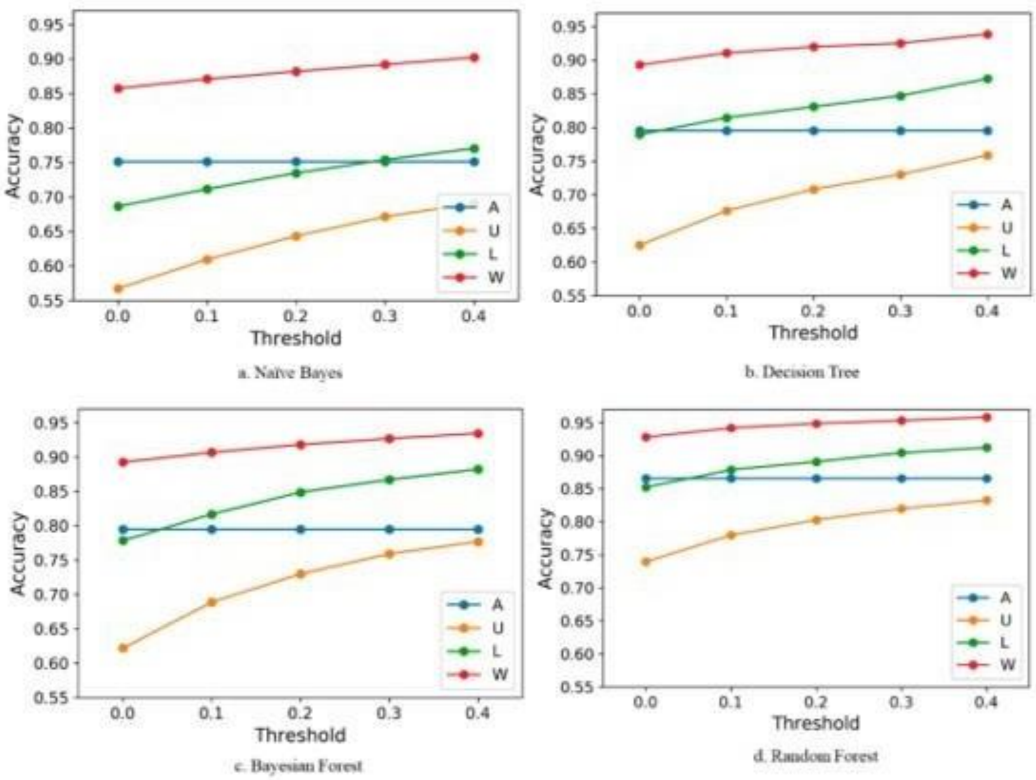
Machine Learning

• Decision Trees (DT) Decision trees are one among the foremost widely implemented supervised learning algorithms and are considered a structured approach for multiclass classification. they're robust and may achieve high accuracy in various tasks while being accountable. the knowledge gained by a choice tree during the training phase is formulated into a hierarchical data structure. This structure is straightforward to interpret even by non-experts. the event of DT usually involves two steps—induction and pruning—in the formation of a tree-like structure. Induction involves tree building, i.e., the formation of nodes and branches of the choice tree [12]

• Naïve Bayes (NB) Classification Naïve Bayes classifier may be a probabilistic machine-learning model, which may be a collection of classification algorithms supported Bayes' Theorem. it's considered fast, efficient, and straightforward to implement. It assumes that the predictive features are mutually independent given the category. during this study, the Bernoulli Naïve Bayes algorithm is employed, where each feature is meant to be a binary-valued variable.[13]

• Random Forest (RF) A random forest is an ensemble of multiple decision trees. it's a well- liked and highly efficient ensemble method for supervised learning algorithms and may be used for both regression and classification. t. it's comparatively a faster method which will identify non-linear patterns in data and may be a good solution to a standard problem with decision trees of overfitting. It works well for both numerical and categorical data.[14]

• .The Bayesian approach starts with aprevious distribution. Subsequently, it estimates a likelihood function for every set of knowledge during a decision tree. Bayesian forest draws the weights of the trees from an exponential distribution and therefore the prediction is an approximate posterior mean.[15]

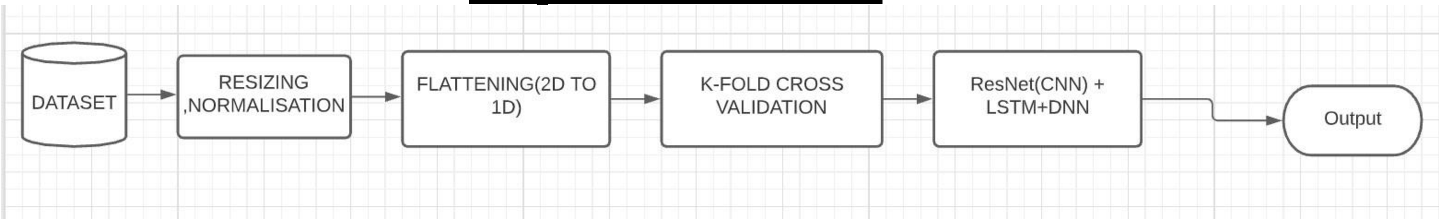


re 6. Accuracies at different thresholds for (a) Naïve Bayes, (b) Decision Trees, (c) Bayesian Forest, (d) Random Forest.

Challenges

1. A Convolutional neural network is significantly slower due to anoperation such as maxpool.
2. If the CNN has several layers then the training process takes a lot of time if the computer doesn't consist of a good GPU.
3. A ConvNet requires a large Dataset to process and train the neural network.

Proposed Architecture



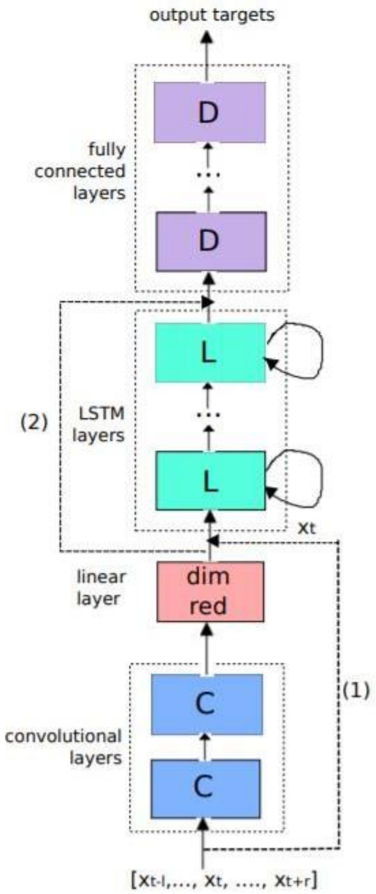
- For pre-processing we'll be using resizing and normalization. In simple words, when multiple attributes are there but attributes have values on different scales, this might cause poor data models while performing dataprocessing operations. in order that they are normalized to bring all the attributes on an equivalent scale.

- For Validation k-fold cross validation. The model is going to be evaluated using 5-fold cross-validation. the worth of k=5 was chosen to supply a baseline for both repeated evaluation and to not be overlarge on require an extended time period. Each test set are going to be 20% of the training dataset, or about12,000 examples, on the brink of the dimensions of the particular test set for this problem.

  The training dataset is shuffled before being split and therefore the sample shuffling is performed whenever in order that any model, we evaluate will havean equivalent train and test datasets in each fold, providing an apples-to- apples comparison

- For classification we'll use CNN with batch normalization just because it produces highest accurate results. We will also use transfer learning to lock the method.

CNN+LSTM+DNN



Specifically, we use 2 convolutional layers, each with 256 feature maps. We use a 9x9 frequency-time filter for the first convolutional layer, followed by a 4x3 filter for the second convolutional layer, and these filters are shared across the entire time-frequency space. Our pooling strategy is to use non-overlapping max pooling, and pooling in frequency only is performed. A pooling size of 3 was used for the first layer, and no pooling was done in the second layer.

After frequency modeling is performed, we next pass the CNN output to LSTM layers, which are appropriate for modeling the signal in time. Following the strategy proposed in, we use 2 LSTM layers, where each LSTM layer has 832 cells, and a 512-unit projection layer for dimensionality reduction.

Finally, after performing frequency and temporal modeling, we pass the output of the LSTM to a few fully connected DNN layers. As shown in, these higher layers are appropriate for producing a higher-order feature representation that is more easily separable into the different classes we want to discriminate. Each fully connected layer has 1,024 hidden units.

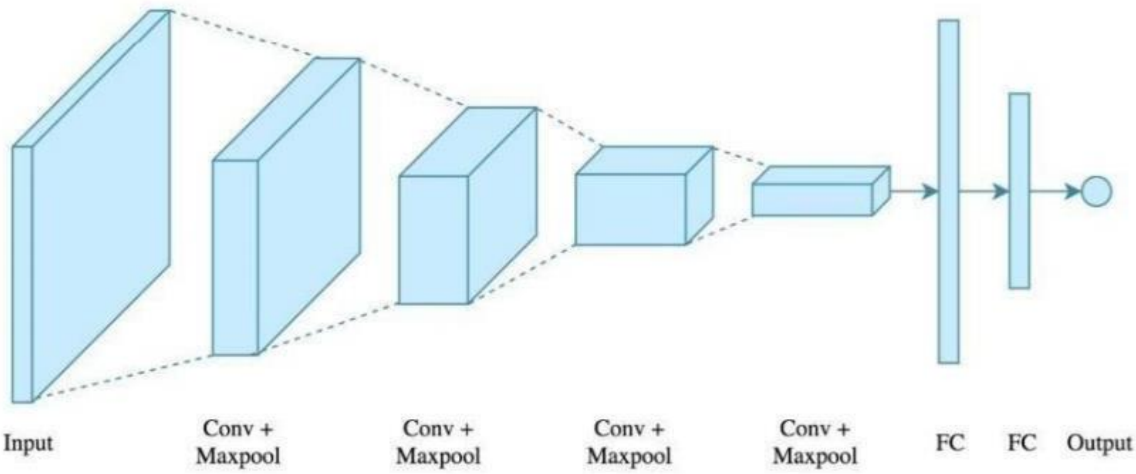


Fig. Cnn Model

CODE

Import the Fashion MNIST dataset

```
[ ] # Import dataset.

    from __future__ import absolute_import, division, print_function, unicode_literals

    # TensorFlow and tf.keras
    import tensorflow as tf
    from tensorflow import keras

    # Import tensorflow's MNIST data handle
    fashion_mnist = keras.datasets.fashion_mnist

    # Load fashion dataset into the set for training & testing
    (train_data, train_label), (test_data, test_label) = fashion_mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
32768/29515 [=================================] - 0s 0us/step
40960/29515 [==========================================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26427392/26421880 [==============================] - 0s 0us/step
26435584/26421880 [==============================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
16384/5148 [===============================================================================================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4423680/4422102 [==============================] - 0s 0us/step
4431872/4422102 [==============================] - 0s 0us/step
```

```
# The original dataset comes with the shape [image_index, 28_, 28_]
# We flatten the data into the shape [image_index, 784 (28_ + 28_)]

#training_data = np.array([img_train.flatten() for img_train in train_data],)
training_data = np.array([data.flatten() for data in train_data], dtype='float32')
# Convert List to array and then Tranpose in order to have an Array of [img_index, 1]
training_label = np.array([train_label]).T
```

Preprocess the data

The data must be preprocessed before training the network. If you inspect the first image in the training set, you will see that the pixel values fall in the range of 0 to 255:

```
[ ]  # normalize the training & testing data
     x_train = training_data[:,:]/255
     y_train = training_label[:, 0]

     x_test = testing_data[:,:]/255
     y_test = testing_label[:, 0]
```

Declaring the layers of CNN:

```
[ ]  # start forming the model. In our case we use an CNN & Dense model in order to solve the problem
     cnn_model = Sequential()
```

```
[ ]  # Fisrt Iteration --> Convulutional with 64 features / 3,3 kernel and Relu activation function
     cnn_model.add(Conv2D(64, 3, 3, input_shape = (28,28,1), activation="relu"))
```

```
[ ]  # MaxPooling layer
     cnn_model.add(MaxPooling2D(pool_size=(2,2)))
```

```
[ ]  # flat images to inputs
     cnn_model.add(Flatten())
```

```
[ ]  # Second Iteration --> Dense of 64 outputs
     cnn_model.add(Dense(units = 64, activation = "relu"))
```

```
[ ]  # The Third Iteration use a Dropout layer to improve the model. Trying to avoid overfitting over the Dense layer
     # Dropout 0.4 ratio
     cnn_model.add(Dropout(rate=0.4))
```

```
[ ]  # Dense of 10 outputs (fashion classes)
     cnn_model.add(Dense(10, activation = "sigmoid"))
```

```
[ ]  # compile model
     cnn_model.compile(loss = 'sparse_categorical_crossentropy',optimizer=Adam(lr=0.001), metrics = ['accuracy'])

     /usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/adam.py:105: UserWarning: The `lr` argument is deprecated, use `learning_r
       super(Adam, self).__init__(name, **kwargs)
```

Training the model:

```
[ ] # define # of epochs
    epochs = 70
```

```
# train the model
cnn_model.fit(X_train,
              y_train,
              batch_size = 400 ,
              epochs = epochs,
              verbose=1,
              validation_data= (X_validate, y_validate))
```

```
Epoch 1/70
120/120 [==============================] - 1s 10ms/step - loss: 0.3702 - accuracy: 0.8650 - val_loss: 0.3327 - val_accuracy: 0.8758
Epoch 2/70
120/120 [==============================] - 1s 9ms/step - loss: 0.3692 - accuracy: 0.8641 - val_loss: 0.3318 - val_accuracy: 0.8762
Epoch 3/70
120/120 [==============================] - 1s 9ms/step - loss: 0.3657 - accuracy: 0.8649 - val_loss: 0.3311 - val_accuracy: 0.8751
Epoch 4/70
```

Evaluation and Prediction:

```
[ ] # Evaluate the model
    evaluation = cnn_model.evaluate(X_test,y_test,batch_size=1)
    # print the evaluation Accuaracy
    print('Test Accuracy : {:.3f}'.format(evaluation[1]))
```

```
10000/10000 [==============================] - 33s 3ms/step - loss: 0.3602 - accuracy: 0.8769
Test Accuracy : 0.877
```

```
[ ] # show the predicted classes (y_label_test) using as input the X_test set
    predict_x = cnn_model.predict(X_test)
    predicted_classes = np.argmax(predict_x, axis=1)
    predicted_classes
```

```
array([9, 2, 1, ..., 8, 1, 5])
```

EXPERIMENTS RESULTS

DATASET

This guide uses the Fashion MNIST dataset which contains 70,000 grayscale images in 10 categories. The images show individual articles of clothing at low resolution (28 by 28 pixels), as seen here:

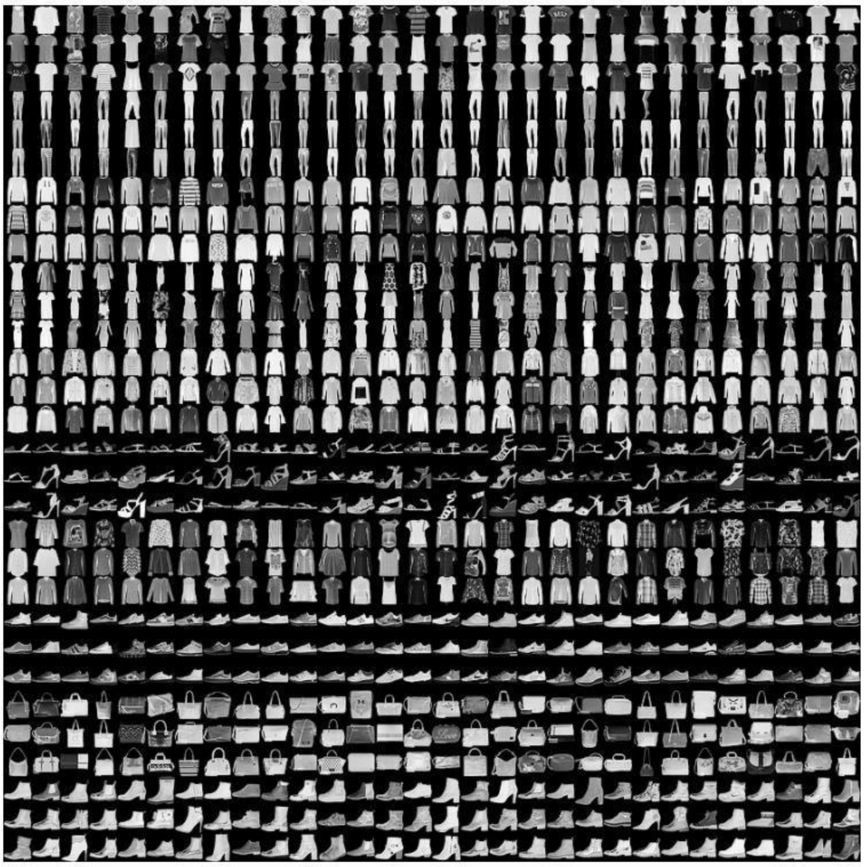


Figure 1. Fashion-MNIST samples (by Zalando, MIT License).

This guide uses Fashion MNIST for variety, and because it's a slightly more challenging problem than regular MNIST. Both datasets are relatively small and are used to verify that an algorithm works as expected. They're good starting points to test and debug code.

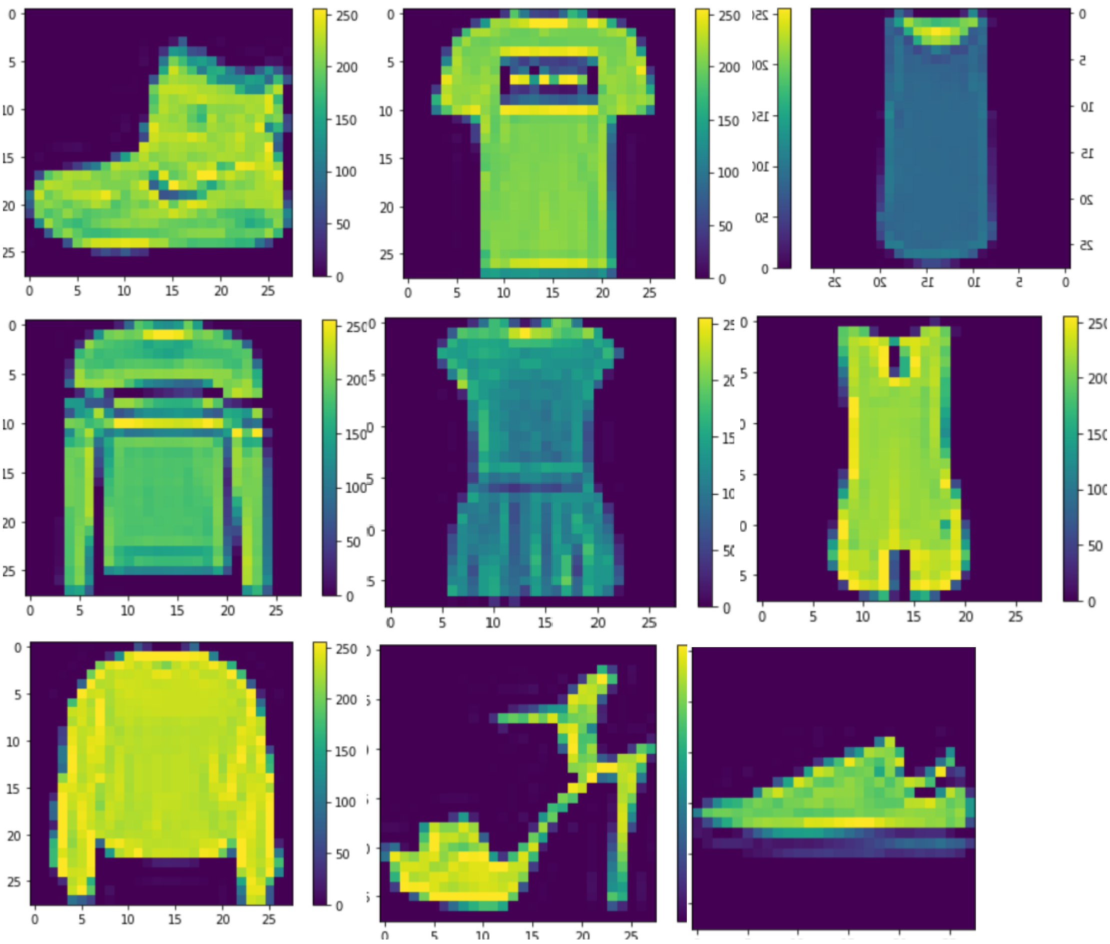
Here, 60,000 images are used to train the network and 10,000 images to evaluate how accurately the network learned to classify images

The images are 28x28 NumPy arrays, with pixel values ranging from 0 to 255. The *labels* are an array of integers, ranging from 0 to 9. These correspond to the *class* of clothing the image represents:

Label	Class
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

Pre-process the data

The data must be pre-processed before training the network.



To verify that the data is in the correct format and that you're ready to build and train the network, let's display the first 25 images from the *training set* and display the class name below each image.

SETUP LAYERS

The first layer in this network, tf.keras.layers. Flatten, transforms the format of the images from a two-dimensional array (of 28 by 28 pixels) to a one-dimensional array (of 28 * 28 = 784 pixels). Thinkof this layer as unstacking rows of pixels in the image and lining them up. This layer has no parameters to learn; it only reformats the data.

TABLE III. DEEP NUERAL NETWORK ARCHITECTURE

Specific Parameter	Value/Type
Layers	7
Input shape	(14, 14, 128)
Epochs	120
Batch-size	64
Optimizer	Adam
Learning Rate	0.001
Loss Function	categorical_crossentropy
Activation Function	ReLU
Dropout	0.5
Last Layer	Softmax

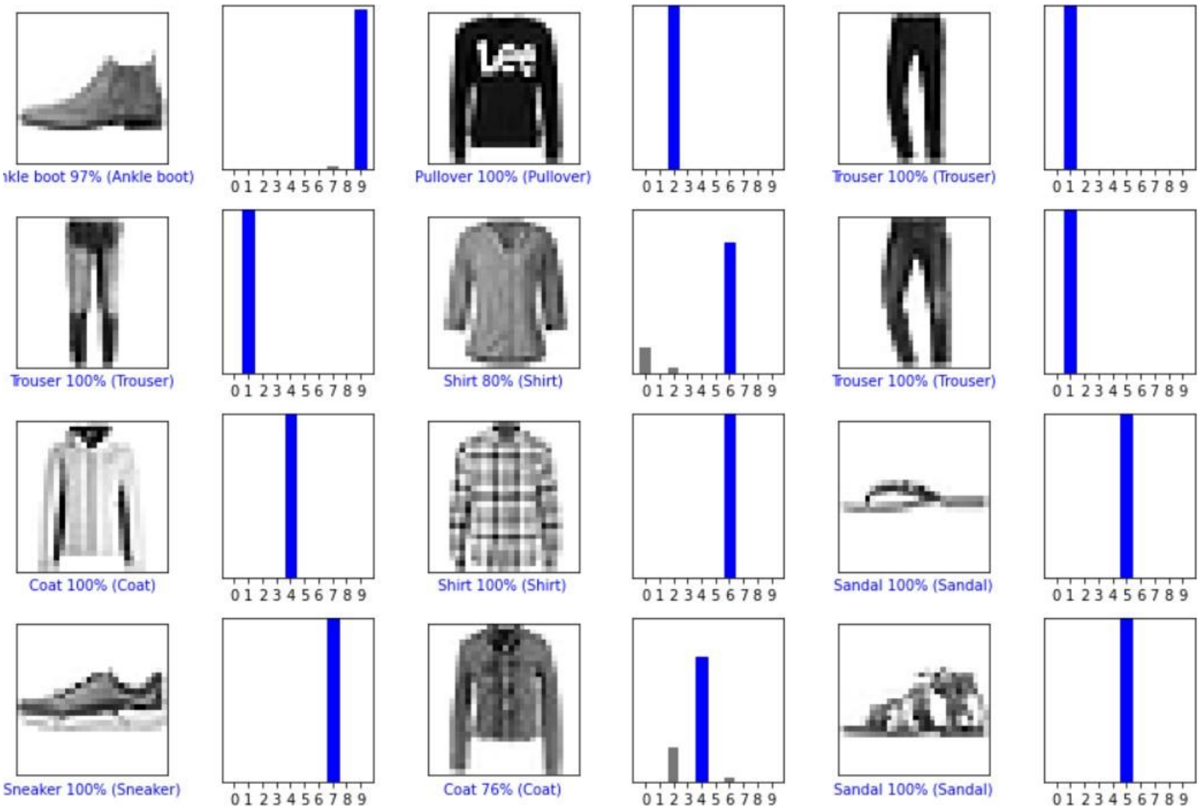
These are added during the model's *compile* step:

- ***Loss function*** **—**This measures how accurate the model is during training. You want to minimize this function to "steer" the model in the right direction.
- *Optimizer* —This is how the model is updated based on the data it sees and its loss function.
- *Metrics* —Used to monitor the training and testing steps. The following example uses *accuracy*, the fraction of the images that are correctly classified.

```
Epoch 55/70
120/120 [==============================] - 1s 9ms/step - loss: 0.2805 - accuracy: 0.8947 - val_loss: 0.3161 - val_accuracy: 0.8868
Epoch 56/70
120/120 [==============================] - 1s 9ms/step - loss: 0.2742 - accuracy: 0.8957 - val_loss: 0.3113 - val_accuracy: 0.8868
Epoch 57/70
120/120 [==============================] - 1s 9ms/step - loss: 0.2721 - accuracy: 0.8987 - val_loss: 0.3150 - val_accuracy: 0.8864
Epoch 58/70
120/120 [==============================] - 1s 9ms/step - loss: 0.2754 - accuracy: 0.8950 - val_loss: 0.3174 - val_accuracy: 0.8870
Epoch 59/70
120/120 [==============================] - 1s 9ms/step - loss: 0.2723 - accuracy: 0.8968 - val_loss: 0.3149 - val_accuracy: 0.8858
Epoch 60/70
120/120 [==============================] - 1s 9ms/step - loss: 0.2713 - accuracy: 0.8983 - val_loss: 0.3117 - val_accuracy: 0.8880
Epoch 61/70
120/120 [==============================] - 1s 10ms/step - loss: 0.2689 - accuracy: 0.8986 - val_loss: 0.3186 - val_accuracy: 0.8861
Epoch 62/70
120/120 [==============================] - 1s 9ms/step - loss: 0.2689 - accuracy: 0.8993 - val_loss: 0.3144 - val_accuracy: 0.8884
Epoch 63/70
120/120 [==============================] - 1s 9ms/step - loss: 0.2683 - accuracy: 0.8997 - val_loss: 0.3200 - val_accuracy: 0.8848
Epoch 64/70
120/120 [==============================] - 1s 9ms/step - loss: 0.2683 - accuracy: 0.8978 - val_loss: 0.3207 - val_accuracy: 0.8864
Epoch 65/70
120/120 [==============================] - 1s 9ms/step - loss: 0.2652 - accuracy: 0.9000 - val_loss: 0.3158 - val_accuracy: 0.8858
Epoch 66/70
120/120 [==============================] - 1s 9ms/step - loss: 0.2657 - accuracy: 0.9000 - val_loss: 0.3162 - val_accuracy: 0.8867
Epoch 67/70
120/120 [==============================] - 1s 9ms/step - loss: 0.2634 - accuracy: 0.9007 - val_loss: 0.3202 - val_accuracy: 0.8879
Epoch 68/70
120/120 [==============================] - 1s 9ms/step - loss: 0.2664 - accuracy: 0.8987 - val_loss: 0.3156 - val_accuracy: 0.8867
Epoch 69/70
120/120 [==============================] - 1s 9ms/step - loss: 0.2608 - accuracy: 0.9009 - val_loss: 0.3217 - val_accuracy: 0.8865
Epoch 70/70
120/120 [==============================] - 1s 9ms/step - loss: 0.2625 - accuracy: 0.9009 - val_loss: 0.3185 - val_accuracy: 0.8872
<keras.callbacks.History at 0x7fde6617c990>
```

Fig. (Test Accuracy)

__MAKE PREDICTION TO GET MOST ACCURATE CLASSIFICATION__



Label	**precision**	**Recall**	**F1-score**
T-shirt/top	90	88	89
Trouser	99	99	99
Pullover	92	91	91
Dress	95	89	92
Coat	92	90	91
Sandal	99	99	99
Shirt	76	85	80
Sneaker	97	98	97
Bag	99	100	99
Ankle boot	99	97	98

TABLE 1 THE PERFORMANCE EVALUATION USING EVALUATION METRICS

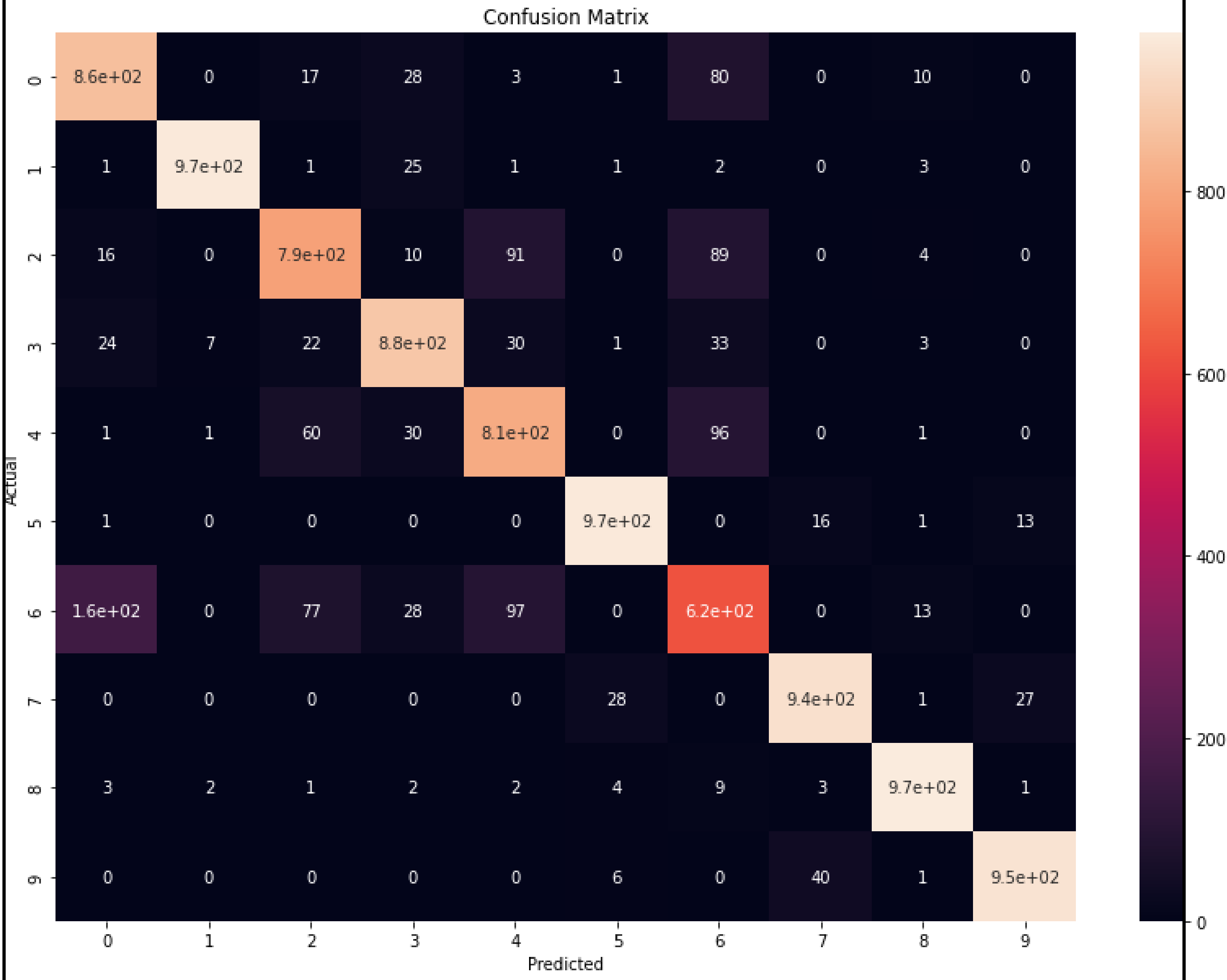
ACCURACY

Acc: 0.877

val_loss: 0.3185

val_acc: 0.8872

Confusion Matrix



I. RESULT AND ANALYSIS

The thing about neural networks is that they can have arbitrarily large complexity. That means that, with enough layers, they can simulate a much larger set of functions. As a result, they can reach a much tighter fit to the data. The drawbacks, however, are that they need a lot more data to learn from and take a lot more time to train. You can potentially get 1 recall and 1 precision with a CNN, but that likely means you did some overfitting.

Random forests can also have arbitrary complexity and their random nature prevents them from overfitting (well, it helps a lot). They aren't great for image data though, since they don't handle huge numbers of features very well.

Nearest Neighbors does beautifully when approximating a function on a compact set and you havea lot of data. Suppose your function is and that the function doesn't vary too wildly. It's pretty smooth. If and you have 100 roughly evenly spaced data points, KNN will do an incredible job. If, you'd need ~10,000 data points to do equally well on the edge cases. You can still do a good job withfar less than that, but my point is that when you have as many dimensions as an image has pixels, it starts to get out of hand. If you have multiple output classes as well, it's even worse.

SVMs are naturally two-class classifiers, so this leads to two options: one-vs- one classifiers or one-vs-all classifiers. Either way, we are talking about a lot of training and wind up with an ensemble of classifiers that don't have any knowledge of each other. That means that their training won't take into account the way they interact, but a CNN will.

<u>The main take-away:</u>

CNNs utilize the spatial info that alternative algorithms do not so as to scale back the number of parameters and overall complexness whereas learning similar info. for tiny issues, this can be inessential and might create CNNs prohibitively dear to coach. For larger issues, the complexness of alternative algorithms really grows quicker, creating CNNs a lot of viable.

CNNs build their own options from raw signal.

Opposed to alternative algorithms that use vector representations wherever each element sometimes makes some sense on its own. Pixels don’t have which means outside the context, however along they'll contain a lot of info concerning the article on an image than a bunch of itsproperties that you just feed into SVM. CNNs will use infinitely robust priors.

This is a primary property of grievous bodily harm pooling layers. smart generalization and unchangeability to native fluctuations make them super scalable, as Conner Davis says in hisanswer.

CNNs believe spatial options.

It will be their strength if the context of the feature is native (a bunch of pixels) and it will we tend to their weakness if the context is distributed (a total of many one-hot encodings is tougher to handle with CNNs, however terribly simple for a choice tree).

CNNs store way more info as parameters than alternative ways.

Trees in RFs will tell you one thing concerning feature importance if you've got interpretable options, however, that’s it. Convnets have a higher manner of utilizing enormous quantity of parameters.

And most significantly, any comparison of CNNs and SVM/RF/KNN on a true task is ridiculous in oneor the opposite manner. Their various functions are therefore different that they don’t even exist within the same dimension. (MNIST dataset is an exception - why?)

Conclusion and Future Work

- **In conclusion, using LSTMs to solve image classification is quite a challenge. After training and testing the dataset through different methods, my CNN model can fit thedataset whose best accuracy is 88.26%.**
- **However, the two main methods which are training pattern reduction and networkpruning cannot contribute appropriately to enhance the performance of my model.**
- **Compared to the relevant paper, my model can still be improved to obtain better results. In the future, for one, I will improve my pruning algorithm to get both accurate and lesscomputational cost model.**
- **For another, I will reduce the training pattern based on the error loss rather thanrandomly training reduction to choose more important training patterns**.

Google Collab Link: https://colab.research.google.com/drive/1iRE-wUdP6piBHdv5atiOKily84VmNGb-?usp=sharing

DISCUSSION:

Convolutional neural networks (CNNs) have accomplished astonishing achievements across a spread of domains, together with medical analysis,associate degreed an increasing interest has emerged in radiology. thoughdeep learning has become a dominant technique in an exceedingly sort ofcomplicated tasks like image classification and object detection, it's not a nostrum.

A tremendous interest in deep learning has emerged in recent years. the foremost established algorithmic rule among varied deep learning models is convolutional neural network (CNN), a category of artificial neural networks that has been a dominant technique in pc vision tasks since the astonishing results were shared on the thing recognition competition called the ImageNet giant Scale Visual Recognition Competition (ILSVRC) in2012.

Deep learning is considered as a recording machine because itdoesn't leave associate degree audit path to clarify its selections. Researchers have projected many techniques in response to the present downside that offer insight into what options are known within the feature maps, referred to as feature visualization, associate degreed what a part ofan input is to blame for the corresponding prediction, referred to as attribution. For feature visualization, Zeiler and Fergus represented some way to envision the feature maps, wherever the primary layers establish tiny native patterns, like edges or circles, and ulterior layers increasingly} mix them into more purposeful structures. For attribution, Zhou et al. projected some way to provide coarse localization maps, referred to as category activation maps (CAMs), that localize the necessary regions in associate degree input used for the prediction. On the opposite hand, its value noting that researchers have recently that detected deep neural networks are at risk of adversarial examples, that are rigorously chosen inputs that cause the network amendment to vary to alter} output while not an apparent change to a personality's.

Although the impact of adversarial examples within the medical domain is unknown, these studies indicate that the manner artificial networks see and predict is completely different from the manner we tend todo.

REFERENCES

1. Hu, Z., Yan, H., and Lin, X. Clothing segmentation using foreground and backgroundestimation based on the constrained Delaunay triangulation.
2. Weakly Supervised Semantic Segmentation with a Multi-Image Model Alexander Vezhnevets Vittorio Ferrari Joachim M. Buhmann.
3. Review of Image Enhancement Techniques,Ruchika Mishra Utkarsh Sharma
4. G. Kumar and P. K. Bhatia, "A Detailed Review of Feature Extraction in Image Processing Systems," 2014 Fourth International Conference on Advanced Computing & Communication Technologies, Rohtak, India, 2014, pp. 5-12, doi: 10.1109/ACCT.2014.74.
5. Li, C., Guo, J., Porikli, F., & Pang, Y. (2018). LightenNet: A Convolutional Neural Network forweakly illuminated image enhancement. Pattern Recognition
6. http://cs231n.stanford.edu/reports/2016/pdfs/286_Report.pdf
7. https://towardsdatascience.com/classifying-fashion-apparel-getting-started-with-computer-vision-271aaf1baf0
8. https://analyticsindiamag.com/fashion-apparel-recognition-using-convolutional-neural-network/
9. https://arxiv.org/pdf/1811.04374.pdf
10. https://link.springer.com/article/10.1007/s00371-020-01885-7

11. https://scholar.google.co.in/scholar?q=fashion+apparel+classification&hl=en&as_sdt=0&as_vis=1&oi=scholart
12. https://www.sciencedirect.com/science/article/abs/pii/S0957417418305992
13. https://ieeexplore.ieee.org/abstract/document/8247638
14. https://www.koreascience.or.kr/article/JAKO201831854885680.page

15. http://vision.stanford.edu/teaching/cs231n/reports/2015/pdfs/BLAO_KJAG_CS231N_FinalPaperFashionClassification.pdf
16. https://ieeexplore.ieee.org/abstract/document/8313740
17. https://openaccess.thecvf.com/content_cvpr_2018/papers/Wang_Attentive_Fashion_Grammar_CVPR_2018_paper.pdf
18. Liu, L.; Su, J.; Zhao, B.; Wang, Q.; Chen, J.; Luo, Y. Towards an Efficient Privacy-PreservingDecision Tree Evaluation Service in the Internet of Things. Symmetry 2020, 12, 103
19. .Fuster-Parra, P.; García-Mas, A.; Cantallops, J.; Ponseti, F.J.; Luo, Y. Ranking Features on Psychological Dynamics of Cooperative Team Work through Bayesian Networks. Symmetry2016
20. . Mao, W.; Wang, F.-Y. Cultural Modeling for Behavior Analysis and Prediction. In Advancesin Intelligence and Security Informatics; Elsevier: Amsterdam, The Netherlands.
21. . Taddy, M.; Chen, C.-S.; Yu, J. Bayesian and Empirical Bayesian Forests.
22. Garment Categorization Using Data Mining Techniques Sheenam Jain and Vijay Kumar
23. Guo, Y., Liu, Y., Bakker, E.M. *et al.* CNN-RNN: alarge-scale hierarchical image classification framework. *Multimed Tools Appl* **77,** 10251–10271 (2018). https://doi.org/10.1007/s11042-017-5443-x
24. Steffens, Alisson & Maria, Anita & Fernandes, Anita & Lyra, Rodrigo & Reis, Valderi & Leithardt, Valderi & Correia, Sérgio & Crocker, Paul & Luis, Rudimar & Dazzi, Scaranto. (2021). Classifying Garments from Fashion-MNIST Dataset Through CNNs. Advances in Science Technology and Engineering Systems Journal. 6. 989-994. 10.25046/aj0601109.

25. Dey, Nilanjan & Mishra, Girish & Kar, Jajnyaseni & Chakraborty, Sayan & Nath, Siddhartha. (2014). A survey of image classification methods and techniques. 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies, ICCICCT 2014. 10.1109/ICCICCT.2014.6993023.
26. Bhatnagar, Shobhit & Ghosal, Deepanway & Kolekar, Maheshkumar. (2017). Classification of fashion article images using convolutional neural networks. 1-6. 10.1109/ICIIP.2017.8313740.
27. Kayed, Mohammed & Anter, Ahmed. (2020). Classification of Garments from Fashion MNIST Dataset Using CNN LeNet-5 Architecture. 238-243. 10.1109/ITCE48509.2020.9047776.