A Novel CNN-LSTM Model Based on Fashion-MNIST image dataset

Project Group 24

Li Xinwei Luo Chengchang Wei Jingjue

***Abstract*—In this project, we want to study whether Long Short-Term Menmory (LSTM) model can work for unsequential image dataset, such as Fashion-MINST dataset. And we also want to investigate how much the performance will be improved if we add LSTM into classic Convolutional Neural Network (CNN) model.**

Keywords: Long Short-Term Memory, Convolutional Neural Network, Fashion-MNIST Dataset, Image Classification, Deep Learning.

I. Introduction & Motivation

In deep learning field, a convolutional neural network (CNN) is a type of artificial neural networks. We most usually implements it to analyze visual imagery [1]. One of the differences between machines and human beings is that human beings can learn things gradually, which depends on the correlation and architecture of human's brain neurons. We should to imitate the architecture of human's brains, in order to let machines to have the capability to learn the behaviors of human beings in some aspects. Therefore, the main architecture of CNN was built based on the architecture of the nerve conduction process of human brain neurons [2]. Current CNN structures have become a baseline method to study machine learning problems (videos, pictures, etc.) [3]. Scholars have created various network models on the basis of the CNN model. Their research have justified to be good at many image correlation problems containing natural picture recognition, handwritten digital recognition, etc [4]. Among them, Reference [5] finished one of the most traditional modelling work used for image recognition recently.

While recurrent neural network (RNN) is becoming a study baseline method for machine learning problems with sequential data (natural language, audio, etc.). Long short-term memory (LSTM) is an recurrent neural network (RNN) structure that was used for modelling temporal sequences in the field of deep learning [6]. One of the main benefits of RNN is that they can connect prior information to the current task, but when the information is too far apart from the current task, RNN model cannot connect long-term information to the current task. However, LSTM is able to learn long-term information and be able to avoid long-term distance problems [7], which is perfect at working with long-term correlation tasks. Therefore, LSTM model can overcome the limitation of the short-term memory of RNN very well. LSTM has a wide use of applications, such as speech recognition, machine translation, etc., due to its structure to learn sequential information in prediction problems.

Integrating RNN and CNN to deal with image data, the main study fields include target detection [9], and image tagging [8], etc. Reference [10] novelly constructed a mixed model ReNet for image classification problems. This model replaces the pooling layer in CNN structure by adding four common RNNs layers. In this project, we want to use the similar idea, and propose our CNN-RNN model for image classification on Fashon-MNIST dataset, in order to see whether our proposed model can improve the entire accuracy, compared with the CNN baseline models.

II. Problem Statement and/or Hypothesis

In this project, we want to study whether the performance will be improved by combining CNN and LSTM model together on Fashion-MNIST dataset. The expected outcome is that our proposed CNN-LSTM model is better than CNN baseline in general. Because we assume that our image data also contains temporal information, which can be extracted by LSTM to improve the performance. We will evaluate our model based on accuracy, recall and AUROC.

III. Prepare Your Report Before Styling

A. Identify the gap

We usually implements CNN model for classification problems, such as image classification [1] and LSTM for regression problems, such as stock price prediciton, natural language processing [11]. There is no theoretical proofs for whether LSTM model can deal with unsequential data. However, based on a special convolution dimensions, CNN will not contain or extract information from other rows. Then, we can treat each row as a independent time step.

B. The current trend for this problem

Reference [11] innovatively constructed a mixed network model CNN-RNN for image classification problems. They treat image data as a two-dimensional wave data. Their proposed model can use the CNN to filter the input wave data, and then the following output will be fed into RNN in order to compute the continuity and dependency features.

C. Importance of the addressing this gap

Although there is no theoretical proofs, this project will try to evaluate whether LSTM can deal with normal image dataset, and provide a instance for following research on the combining model problems for unsequential dataset.

IV. Literature Review & Project Objectives

A. Literature Review

We perform our method on Fashion-MNIST dataset rather than the original MNIST dataset due to the latter one has a lot of handwritten digits, which is usually regarded as a benchmark and the first choice tried by researchers. Therefore, the reasons why we don't use MNIST are as follows:

- *MNIST is too easy*. With the development of machine learning, the accuracy of MNIST can achieve 99.7% by convolutional nets and 97% by classic machine learning algorithms.
- *MNIST is overused*. As we mentioned above, MNIST dataset is the first choice tried by researchers.
- *MNIST can not represent modern CV tasks*. The idea came from François Chollet, who is a deep learning expert.

1) Fashion-MNIST Dataset: Fashion-MNIST dataset, based on the assortment on Zalando's website, contains 28x28 grayscale images of 70000 unique fashion products, coming from different gender groups: men, women, kids and neutral. All 70000 images are chosen from front look thumbnail images, which are classified into 10 classes with 7000 images per class: T-Shirt/Top, Trouser, Pullover, Dress, Coat, Sandals, Shirt, Sneaker, Bag, and Ankle boots. Fig.1 from reference [17] shows the classification and the pictures of each class.

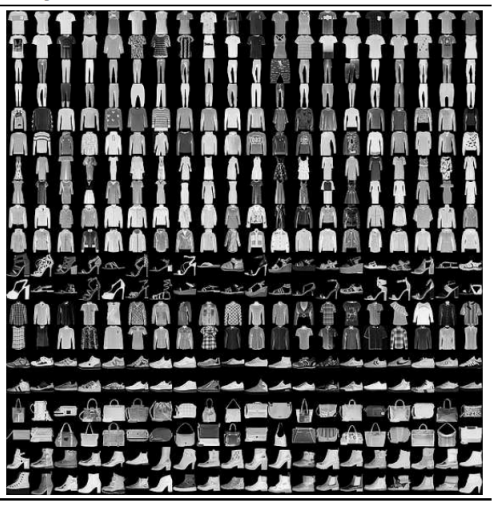
Label	Description	Examples
0	T-Shirt/Top	
1	Trouser	
2	Pullover	
3	Dress	
4	Coat	
5	Sandals	
6	Shirt	
7	Sneaker	
8	Bag	
9	Ankle boots	

Fig. 1. Classes of Fashion-MNIST dataset

2) CNN: Convolutional neural network (CNN) is a kind of multi-layer neural network. Y. LeCun in reference [12] proposed the architecture of LeNet-5 (one kind of CNN), which is shown in Fig.2.

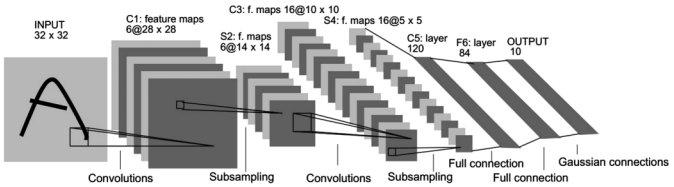


Fig. 2. Construction of LeNet-5

A CNN usually consists of three layer types: Convolutional layer, Pooling (Subsampling) layer, and Fully-Connected layer:

- *Convolutional Layer*. This is a core part of CNN, aiming to abstract features from the input images [18]. The convolutional layer contains multiple feature maps, which are extracted from the input images by convolution filters (kernels). Each filter is a weight matrix with size of 3x3 or 5x5 (usually odd dimensions), and since there are several layers in a CNN, by moving the filter with a given step parameter, we can obtain local characteristics of different positions from the former layer. Equation (1) and (2) here show the formulas when choosing filter number as 1 and step parameter as 1, where $a^{[l-1]}$ is the output of $(l-1)$ layer ($a^{[0]}$ is the original input images), $W^{[l]}$ is the filter of l layer, $b^{[l]}$ is the bias of l layer. After the filter step, the ouputs are passed into a nonlinear activation function. Common activation functions include sigmoid, tanh, and ReLu. As shown in equation (3), g is an activation function, obtaining $a^{[l]}$ as the input of next layer.

$$z^{[1]} = W^{[1]}a^{[0]} + b^{[1]} \tag{1}$$

$$z^{[l]} = W^{[l]}a^{[l-1]} + b^{[l]} \tag{2}$$

$$a^{[l]} = g(z^{[l]}) \tag{3}$$

- *Pooling Layer*. The pooling layer is usually placed between two convolutional layers [19]. whose main purpose is to gradually reduce the spatial size of the data, so that we can reduce the number of parameters in the network, computational cost and effectively control overfitting. The most common way is using a 2x2 filter with a step size of 2 to reduce the dimensions of feature maps. The typical pooling operations are average pooling (take average of the filter space) and max pooling (take the max value of filter space).
- *Fully-connected Layer*. After the process of several convolutional and pooling layers, we have obtained the feature maps of the input images. Then all the neurons in the former layer are connected with each single neuron in fully-connected layer [20]. Finally, the last fully-connected layer is followed by an output layer, and the outputs are used to do classfication tasks, where softmax regression is commonly used.

Recalling the LeNet-5 in figure 2, it consists of 2 convolutional layers and 2 pooling (subsampling) layers, and 2 fully-connected layers, which are the basic components of CNN.

3) LSTM: Recurrent Neural Network (RNN) is initially designed for capturing temporal features of sequential data in David Rumelhart's work in 1986 [21]. A classic RNN is comprised of an input layer, hidden layers and an output layer. The past time-steps inputs more or less influence the current time steps output through the chain-like architecture. However, Hochreiter and and Bengio have discussed that RNNs can't handle "long-term dependencies", widely known as "the problem of vanishing gradients" later, revealing in RNN model the former time-step t's input has little weight to predict the latter time-step t''s output, if $t' - t$ is large [22] [23]. Long-Short Term Memory (LSTM) model was born for addressing this issue.

LSTM model is a derivative of RNN, which shares the same architecture except replacing hidden layer chunks with memory cells, showed in Fig. 3. This special memory cell helps LSTM to keep long-term memory and decide when is suitable to forget it. The memory cell for timestep t consists of three gates: forget gate $\Gamma_f^{<t>}$, update gate $\Gamma_u^{<t>}$ sometimes also called input gate, and output gate $\Gamma_o^{<t>}$. In addition, $x^{<t>}, a^{<t>}, y^{<t>}, c^{<t>}$, and $\tilde{c}^{<t>}$ are the input, activation, output, cell state, and candidate cell state, respectively.

For each timestep t, the memory cell contains:

$$\Gamma_f^{<t>} = sigmoid(W_f[a^{<t-1>}, x^{<t>}] + b_f) \quad (4)$$

$$\Gamma_u^{<t>} = sigmoid(W_u[a^{<t-1>}, x^{<t>}] + b_u) \quad (5)$$

$$\tilde{c}^{<t>} = tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c) \quad (6)$$

$$c^{<t>} = \Gamma_f^{<t>} \circ c^{<t-1>} + \Gamma_u^{<t>} \circ \tilde{c}^{<t>} \quad (7)$$

$$\Gamma_o^{<t>} = sigmoid(W_o[a^{<t-1>}, x^{<t>}] + b_o) \quad (8)$$

$$a^{<t>} = \Gamma_o^{<t>} \circ tanh(c^{<t>}) \quad (9)$$

, where W, b are parameters, and $\circ$ is element-wise product.By iterating the above procedure at each time-step, we finally get the full LSTM architecture in Fig. 4 [24] [26].

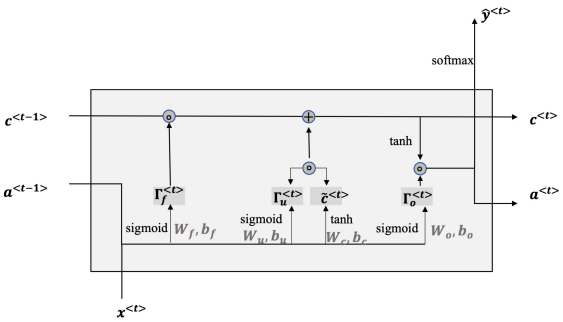


Fig. 3. Memory cell components

There are some adaptations of LSTM to make it potent for a specific type of sequential data. For instance, Bidirectional LSTM (BLSTM) allows $X^{<t>}$ to backward affect $y^{<t-m>}$ for $m > 0$ [26]. Furthermore, adding a Conditional Random Field (CRF) layer after the BLSTM memory cell layer (BLSTM-CRF) can receive a more robust and accurate sequence tagging result by exploiting the dependence between

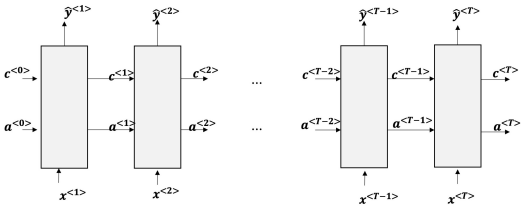


Fig. 4. LSTM architecture

output sequences [25]. Meanwhile, Sutskever Neural Machine Translation (NMT) Model was created to use one layer of LSTM as an encoder to translate input into a fixed-length internal representation and let another LSTM layer decode the vectors into output values in order to deal with the unbalanced input and output length in NLP tasks. Later, Encoder-Decoder LSTM augmented with attention mechanism further removed the limitation of fixed-length of internal representation vector [27]. All of these developments and revolutions keep LSTM holding first place for traditional sequential data field like language translation, audio recognition, etc.

LSTM also has strong potential for the image classification task. Lately, Sequencer, a deep LSTM model with self-attention, can achieve state-of-the-art performance among diverse benchmark datasets compared to CNNs and even Vision Transformers [28].

4) CNN-LSTM: A combined CNN-LSTM network uses the advantage of both CNN and LSTM model, where CNN layers extract the features from images, and LSTM layers extract features from the results of CNN layers to provide sequence prediction [14]. For instance, considering a image dataset, the CNN layer extract features in time series after pre-processing, which is used as input for LSTM layer where the classicication task is done [15]. The Fig. 5 from reference [13] shows the basic construction of CNN-LSTM network.

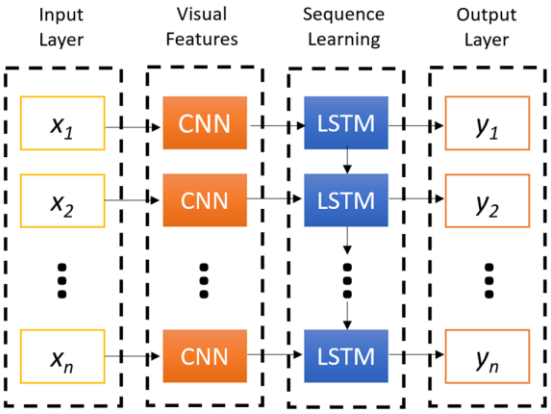


Fig. 5. The basic construction of CNN-LSTM network

B. Project Objectives

In this project, experiments are designed to answer the following questions and achieve the following target:

- What's the effect of the number of convolutional layers on the proposed CNN - LSTM hybrid model?
  We will train two baseline model without LSTM layers whose Convolutional layer is 16 and 32 and two developed model with LSTM whose convolutional layer staying the same to get the conclusion.
- With the same number of CNN layers, what the accuracy and computional complexity of different types of LSTM layers, bidirectional V.S. non?
- Our final hybrid model aims at having accuracy higher than the only external baseline model in [32], whose accuracy is 88.46%.

V. Assumptions & equirement

A. Underlying Assumptions

In our project, our dataset is image. We assume that every row pixel of the image is a time step, which therefore contains time information. For instance, considering a dataset with each image is 28 pixels (rows) by 28 pixels (cols). We can treat each image as a sequence of data, that is, the first row is the first step; The second row is the second step and so on. Therefore, we can consider every image as a sequence of row pixels, then we can fed the data into a LSTM network.

B. Requirements

- Computing resources: Since we are doing a deep learning project, we are in great need of GPU. Because the training process requires a lots of computing resources. However, we only have our own personal computer. Therefore, a potential solution is that we will try to use Google Colaboratory or Amazon Web Services. But, we need some time to learn how to use them.
- Pre-knowledge learning: we also need to learn how to use pytorch, in order to design a feasible model for this project.

VI. Contributions & Success Measures

A. Contributions

This research aims at applying CNN-LSTM hybrid model to deal with image classification tasks. The contribution of this research can be summarized as below:

- Update the best performance of the CNN-LSTM method on Fashion-MNIST image dataset. Since there is only single research on CNN-LSTM model application on Fashion-MNIST image dataset, whose structure is relatively simple [32], offering us chance to build a more subtle model has more delicate LSTM layers and higher the accuracy.
- The performance will compare with other related methods, which offers a good oppurtunity to dive into LSTM and CNN's features.

B. Success Measures

The 10-fold Cross Validation test result will be compared by four metrics Accuracy, F1 scores, AUC and t-SNE visualization.

TABLE I
Confusion Matrix of multi-class m

		Predicted condition			
Actual condition		**P1**	**P2**	$\cdots$	**Pm**
	1	T_1	F_{12}	$\cdots$	F_{1m}
	2	F_{21}	T_2	$\cdots$	F_{2m}
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	m	F_{m1}	F_{m2}	$\cdots$	T_m

1) Accuracy: For each class $i = 1, 2, \cdots, m$:

$$TP_i = T_i \tag{10}$$

$$TN_i = \Sigma_{j \neq i} T_i + \Sigma_{j \neq k \neq i} F_{jk} \tag{11}$$

$$FP_i = \Sigma_{j \neq i} F_{ij} \tag{12}$$

$$FN_i = \Sigma_{j \neq i} F_{ji} \tag{13}$$

$$Accuracy_i = \frac{TP_i}{TP_i + FN_i} \tag{14}$$

$$Recall_i = \frac{TP_i}{TP_i + FP_i} \tag{15}$$

$Accuracy_i$ and $Recall_i$ are class based mean accuracy and recall [30].

In total,

$$TP = \Sigma_i TP_i \tag{16}$$

$$FP = \Sigma_i FP_i \tag{17}$$

$$FN = \Sigma_i FN_i \tag{18}$$

$$Accuracy = Recall = \frac{TP}{TP + FN} \tag{19}$$

The Accuracy of the total multi-class model inherits a similar idea from the two-class accuracy, only if the Precision, Recall, and Accuracy are the same in the multi-class situation.This metric will be used in both external related model comparison with our models and internal comparison among our models, While the last two metrics, AUC and t-SNEs, are only used for internal comparison [29].

2) F1 score:

$$F1 = \frac{2Accuracy \times Recall}{Accuracy + Recall} = Accuracy = Recall \tag{20}$$

Micro F1 takes the same value with total Accuracy in multi-class classification.

3) AUC: Receiver Operating Characteristics (ROC) Curve is a curve of True Positive Rate (TP Rate) against False Positive Rate (FP Rate), which means the probability of the model to discriminate between right cases and wrong cases. The bigger Area Under ROC (AUC) is, the better classifier the model is. Normally, plotting AUCs together will easily get the conclusion of which model is the best. Furthermore, we can plot AUC stratified by class to compare their performances.

4) t-SNEs: t-Distributed Stochastic Neighbor Embedding (t-SNE) is a powerful visual method illustrating model's capacity of classification. It offers an outlook of model's performance and reveals individual level information [31].

VII. Project Plan

First, we will learn the basic knowledge in deep learning field, such as activation functions, stochastic gradient descent via online resources. Then, we need to be familiar with the pytorch API. Because we will implement our model and evaluations using pytorch. Then, we will build two CNN models as baseline with different dimensions. And we will add a same LSTM layer into these two models, in order to check whether the performance will be improved after adding a LSTM. Also, we will try two LSTM layers with different dimensions. Therefore, we will have four models in total, and we will compare these models by accuracy, recall based on confusion matrix. Finally, we will pick up the best model and draw the AUROC for the worst label prediction of the best model.

References

[1] Yamashita, Rikiya, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. "Convolutional neural networks: an overview and application in radiology." Insights into imaging 9, no. 4 (2018): 611-629.

[2] Sahinbas, Kevser, and Ferhat Ozgur Catak. "Transfer learning-based convolutional neural network for COVID-19 detection with X-ray images." In Data Science for COVID-19, pp. 451-466. Academic Press, 2021.

[3] Fukushima, Kunihiko, Sei Miyake, and Takayuki Ito. "Neocognitron: A neural network model for a mechanism of visual pattern recognition." IEEE transactions on systems, man, and cybernetics 5 (1983): 826-834.

[4] Yin, Qiwei, Ruixun Zhang, and XiuLi Shao. "CNN and RNN mixed model for image classification." In MATEC web of conferences, vol. 277, p. 02001. EDP Sciences, 2019.

[5] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[6] Sak, Hasim, Andrew W. Senior, and Françoise Beaufays. "Long short-term memory recurrent neural network architectures for large scale acoustic modeling." (2014).

[7] Salehinejad, Hojjat, Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee. "Recent advances in recurrent neural networks." arXiv preprint arXiv:1801.01078 (2017).

[8] Mao, Junhua, Wei Xu, Yi Yang, Jiang Wang, and Alan L. Yuille. "Explain images with multimodal recurrent neural networks." arXiv preprint arXiv:1410.1090 (2014).

[9] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems 28 (2015).

[10] Visin, Francesco, Kyle Kastner, Kyunghyun Cho, Matteo Matteucci, Aaron Courville, and Yoshua Bengio. "Renet: A recurrent neural network based alternative to convolutional networks." arXiv preprint arXiv:1505.00393 (2015).

[11] Yin, Qiwei, Ruixun Zhang, and XiuLi Shao. "CNN and RNN mixed model for image classification." In MATEC web of conferences, vol. 277, p. 02001. EDP Sciences, 2019.

[12] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[13] Tasdelen, A., and Sen, B. (2021). A hybrid CNN-LSTM model for pre-miRNA classification. Scientific reports, 11(1), 1-9.

[14] Islam, M. Z., Islam, M. M., and Asraf, A. (2020). A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. Informatics in medicine unlocked, 20, 100412.

[15] Agga, A., Abbou, A., Labbadi, M., El Houm, Y., and Ali, I. H. O. (2022). CNN-LSTM: An efficient hybrid deep learning architecture for predicting short-term photovoltaic power production. Electric Power Systems Research, 208, 107908.

[16] Li, Y., He, Y., and Zhang, M. (2020). Prediction of Chinese energy structure based on convolutional neural network-long short-term memory (CNN-LSTM). Energy Science & Engineering, 8(8), 2680-2689.

[17] Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747.

[18] Guo, T., Dong, J., Li, H., and Gao, Y. (2017, March). Simple convolutional neural network on image classification. In 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA) (pp. 721-724). IEEE.

[19] Wang, W., Yang, Y., Wang, X., Wang, W., and Li, J. (2019). Development of convolutional neural network and its application in image classification: a survey. Optical Engineering, 58(4), 040901.

[20] Sharma, N., Jain, V., and Mishra, A. (2018). An analysis of convolutional neural networks for image classification. Procedia computer science, 132, 377-384.

[21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.

[22] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Juergen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In S. C. Kremer and J. F. Kolen, eds., A Field Guide to Dynamical Recurrent Neural Networks. IEEE press, 2001.

[23] Y. Bengio, P. Simard and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," in IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 157-166, March 1994, doi: 10.1109/72.279181.

[24] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[25] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF Models for Sequence Tagging." arXiv, Aug. 09, 2015. Accessed: Sep. 03, 2022. [Online]. Available: http://arxiv.org/abs/1508.01991

[26] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," Neural Networks, vol. 18, no. 5–6, pp. 602–610, Jul. 2005, doi: 10.1016/j.neunet.2005.06.042.

[27] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks." arXiv, Dec. 14, 2014. Accessed: Sep. 03, 2022. [Online]. Available: http://arxiv.org/abs/1409.3215

[28] Y. Tatsunami and M. Taki, "Sequencer: Deep LSTM for Image Classification." arXiv, May 17, 2022. Accessed: Sep. 03, 2022. [Online]. Available: http://arxiv.org/abs/2205.01972

[29] F. O. Ozkok and M. Celik, "A hybrid CNN-LSTM model for high resolution melting curve classification," Biomedical Signal Processing and Control, vol. 71, p. 103168, Jan. 2022, doi: 10.1016/j.bspc.2021.103168.

[30] Y. Hua, L. Mou, and X. X. Zhu, "Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional LSTM network for multi-label aerial image classification," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 149, pp. 188–199, Mar. 2019, doi: 10.1016/j.isprsjprs.2019.01.015.

[31] Maaten, L.V.D., Hinton, G., 2008. Visualizing data using t-sne. J. Mach. Learn. Res. 9 (Nov), 2579–2605.

[32] K. Zhang, "LSTM: An Image Classification Model Based on Fashion-MNIST Dataset," p. 7.