**Alireza Mahinparvar**

**ISE 244 – Spring 2023**

**LS-SVM algorithm in handwriting recognition**

**Problem definition:**

As we know in today’s world the usage of Machine Learning algorithms and Artificial Intelligence is increasing as we see lots of products using these technologies. One of the crucial issues and problems that tech companies faced during the past decade was the issue related to handwriting recognition. As we know these days many people, companies and organizations use DocuSign as digital signatures for document processing or post services uses automated mail sorting. These issues are all related to handwriting recognition and for the past years’ scientists have tried to use machine learning algorithms to solve issues related to this field. The reason that this field is important is that by improving handwriting recognition companies and people can save time in them for example post service companies can recognize zip code for postal mail sorting, banks can process check amounts easily, companies like turbo tax can fill the documents by handwriting recognition techniques. Handwriting recognition paves the path to optimal storage. Documents, many files, contracts, and personal records include handwritten information, such as original signatures or notes, that can be converted into electronic text with handwritten text recognition technologies.

Lot of scientists are working to problem related to handwriting recognition for example authors in [3] proposed the use of Deep learning Convolutional Neural Networks (CNNs) as the core architecture for handwriting recognition systems. CNNs are mostly used and well known for image processing tasks and excel at capturing patterns in visual data. Authors believe that in comparison to other models this deep learning technique improves accuracy and the ability to learn complex patterns and overall, it gives system advantage compared to other models.

In another paper [4] author proposed the usage of LSTM networks, a type of recurrent neural network (RNN), as core architecture for online handwriting recognition. LSTM networks are well known for sequence modeling tasks and can capture long-term dependencies in sequential data. The paper focuses on the need for fast recognition performance in online handwriting applications.

In another paper [5] we see authors focused on using handwriting as the modality of communication for between in the brain-computer interface. This paper explores the feasibility of decoding neural network signals that are associated with handwriting and translating them into text. They believe that this technology can significantly improve the quality of life for individuals with motor impairments by allowing them to communicate more effectively and efficiently.

Finally in paper [2] which we are going to spend rest of this project on talks about handwriting recognition in the context of model selection for the LS-SVM algorithm. As we know Model selection is a crucial step in creating ML models because it entails choosing the right hyperparameters to ensure the model performs at its best. The model selection process for the LS-SVM in this instance can entail selecting the kernel function, regularization parameter, and other model performance-related factors. In addition to this paper, we see that authors in paper [1] propose a framework that automates the process of model selection  
and hyperparameter optimization for a wide range of classification algorithms including support  
vector machine (SVMs). This is the framework that can be useful for optimizing the LS-SVM

**Project Objectives:**

The hypothetical project entails creating a machine learning model for handwriting  
recognition-based automated mail sorting. This project aims to automate mail sorting  
using machine learning techniques to increase its effectiveness and accuracy. In this project we are planning to compare results from LS-SVM model which described in [2] to other famous deep learning models that have that described in [3] and [4] which are CNNs and RNNs. Results from this comparison allow us to have a better understanding of which algorithms work better for automated mail sorting.

The project can result in increased customer satisfaction as well as cost savings for the  
company by increasing the efficiency and accuracy of mail sorting. Errors in mail sorting  
that could result in delayed or lost mail are among the potential risks of damage. The  
possibility for unexpected repercussions or biases in the machine learning model, which  
could produce biased outputs, could also be considered as a downstream influence.

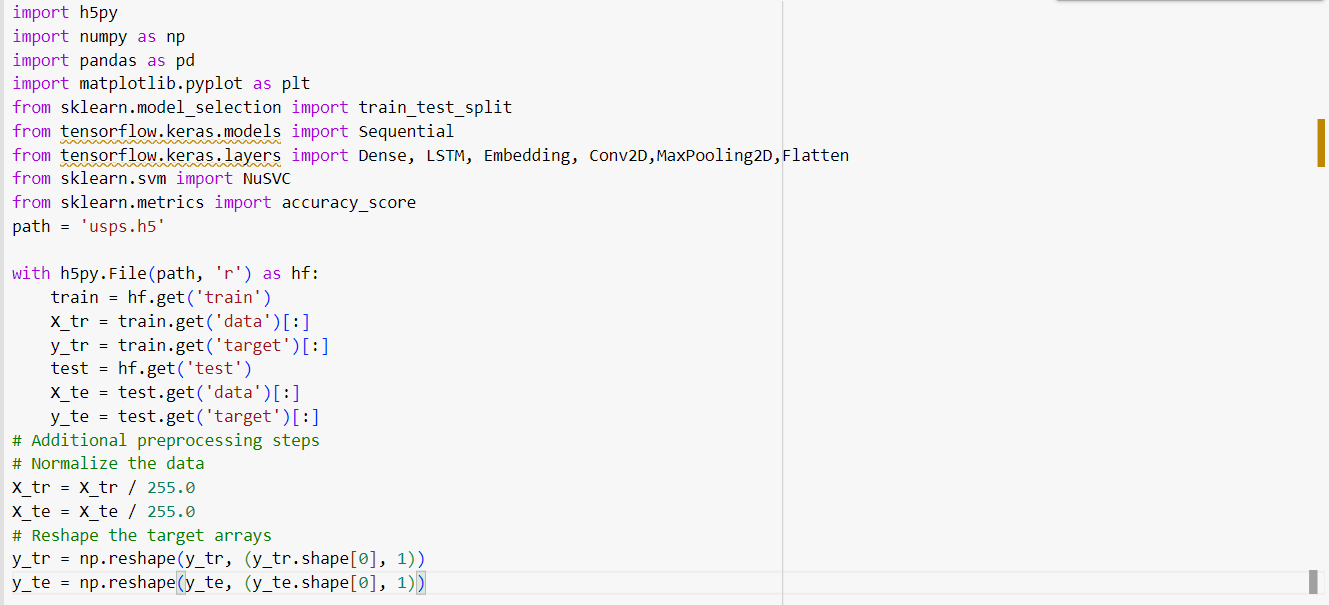
While our plan is to use machine learning techniques to come up with LS-SVM on a dataset and compare results out of it with CNNs and RNNs, we may face difficulties and limitations in this process. The main issue can be size and diversity of datasets along with data labeling. Looking for datasets that offer diverse and sufficiently large collection of handwriting samples can be hard also we need to make sure that data in the dataset is labeled reliably for handwriting samples. Finding the right dataset for our application specifically which is mail sorting. It’s crucial and critical to have datasets that are relevant, have good quality and compatible with our project goals. Additionally applying some additional processes like normalization, augmentation techniques may be difficult as well and that can be directly related to dataset.

Before we start working on datasets and analyses it’s better to have a good understanding of the method that we want to use and methods we want to compare through this project. The first and main algorithm that are planning to implement on our dataset is LS-SVM. LS-SVM stands for Least Squares Support Vector Machine and its variant of support vector machine that combines least squares regression and SVM classification and its useful for solving regression and classification problems. This algorithm can be used to learn mapping from input feature (such as pixel intensities in an image) to output labels (e.g., the corresponding character or digit). By training this model on our desired dataset it can recognize unseen handwritings. Another model that we use to compare our LS-SVM with it is CNN or Convolutional Neural Network, CNN is a type of deep learning model that is effective for image processing and analyses which includes handwriting recognition. CNN is designed to automatically learn and extract features from images and make them well suited for tasks that involve recognizing patterns. The way that CNN helps for handwriting recognition is that it extract features from input image and uses convolutional layer along with pooling layer. Throughout these layers of pooling and convolution, lower layers capture low-level features such as edges and textures, while higher level captures complex patterns and features. CNNS often ends with one of the more fully connected layers that allows to capture dependencies among extracted features. RNN is another type of neural network that we will use to compare our model’s tool. This model is designed to effectively process sequential data by maintaining an internal memory that captures information from previous steps in sequence. By leveraging sequential nature of handwriting data and the ability to capture dependencies across different pen strokes, RNNs can effectively learn and recognize handwriting patterns. They can capture the temporal dynamics and context within the sequence, allowing for accurate recognition and understanding of handwriting text.

**Analysis:**

For our projects we decided to choose USPS dataset which is available at : [USPS dataset | Kaggle](https://www.kaggle.com/datasets/bistaumanga/usps-dataset?resource=download). The dataset has 7291 trains and 2007 test images. The images are 16\*16 grayscale pixels.

The dataset is given in [hdf5 file format](https://support.hdfgroup.org/HDF5/), the hdf5 file has two groups train and test and each group has two datasets: data and target.



Figuire1

As we see in figure 1, to read data we need to apply the following code in it to get train data and target data. Also, the preprocessing and normalization along with reshape applied to the data. After following step it’s the time to put data into our methods for testing:

In the first code, we start testing the data using the RNN method. The code starts by splitting the data into a training and validation set using the train\_test\_split function. The data is then reformatted to match the input format required by the RNN model. The RNN model is defined by the Keras sequential class and consists of a 128-unit LSTM layer followed by a 10-unit density layer and a softmax activation function. The model is assembled with Adam optimizer and sparse\_categories\_crossentropy losses. The RNN model is trained on the training data using a fitting function and validated on the validation set. After training, the performance of the model is evaluated with the test set and the evaluation function of the validation set, and the results are printed. The exact results are stored in a data frame and displayed as a table. Finally, a bar graph is created to visualize the accuracy of the RNN model on the validation and test set. Results from figure 2 on RNN model shows us that this model gave us accuracy of 56% on validation set and 54% on test set.



Figure 2

The data in the second part of the code is reformatted to match the input format required by the CNN model. The input data is reformatted using np. reformat so that it has dimensions of (-1, 16, 16, 1), where -1 is the number of samples, 16 is the height and width of the image, and 1 is the number of channels (grayscale). The data is then split into a training and validation set using the train\_test\_split command. The CNN model is defined using a sequential class and consists of convolutional layers with 32 and 64 filters, respectively, followed by a maximum pooling layer. The smoothed output is combined with two density layers with ReLU and softmax activation functions of 64 and 10 units, respectively. The model is built with Adam Optimizer and sparse\_category\_crossentropy losses. The CNN model is trained on the training data using a fit function and validated on the validation set. The performance of the model is evaluated with the test set and the evaluation function of the validation set, and the results are printed. Accuracy results are stored in a DataFrame and displayed as a table. Finally, a bar chart is created to visualize the accuracies of the CNN model on the validation and test set. From figure 3 we see that accuracy of validation 79% and accuracy of test set is 72%

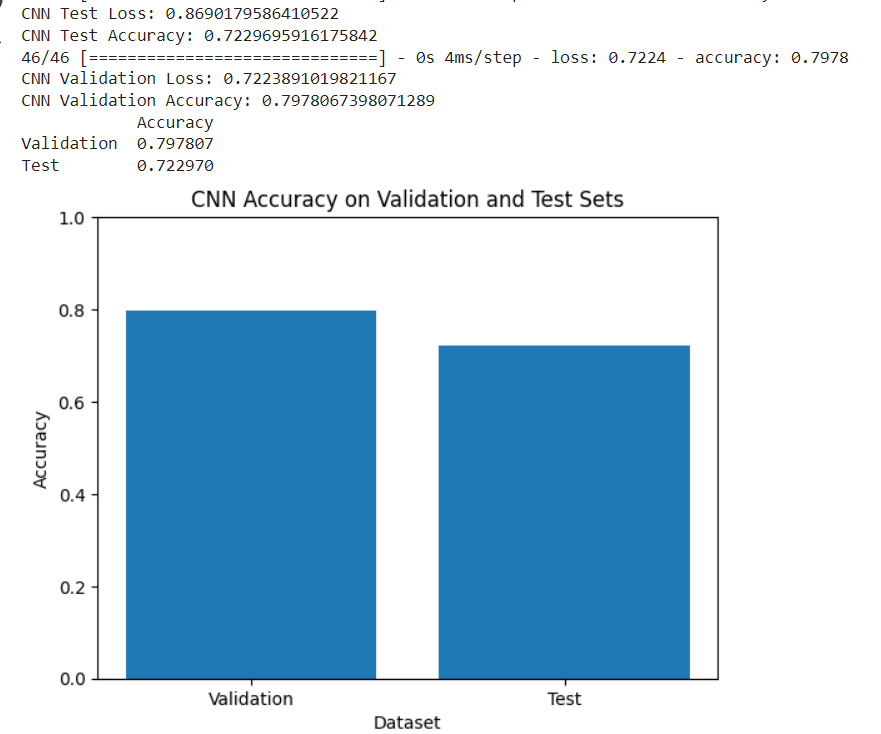


Figure 3

For the third part of the code, we modify the input data to fit the LS-SVM model. The data is reformatted using np. reformat so that its dimensions are (number of samples, -1), where -1 means that the remaining dimensions are decided automatically. The reformulated data is then split into training and validation sets using the train\_test\_split function. In this case, an LS-SVM model represented by NuSVC is defined. The model is trained using the training data by calling the appropriate method. Then, the model predicts the labels of the validation and test set using the prediction method. Accuracy scores are calculated by comparing predicted tags with actual tags using the accuracy\_score function. Validation and test accuracy are printed and stored in a DataFrame for display. A bar graph is created to visualize the accuracy of the LS-SVM model on the validation and test set. This code can be included in the analysis section of the report to describe the implementation, evaluation, and accuracy of the LS-SVM model on the validation and test set. As we see in figure 4 the validation accuracy is 94% and test accuracy is 89%

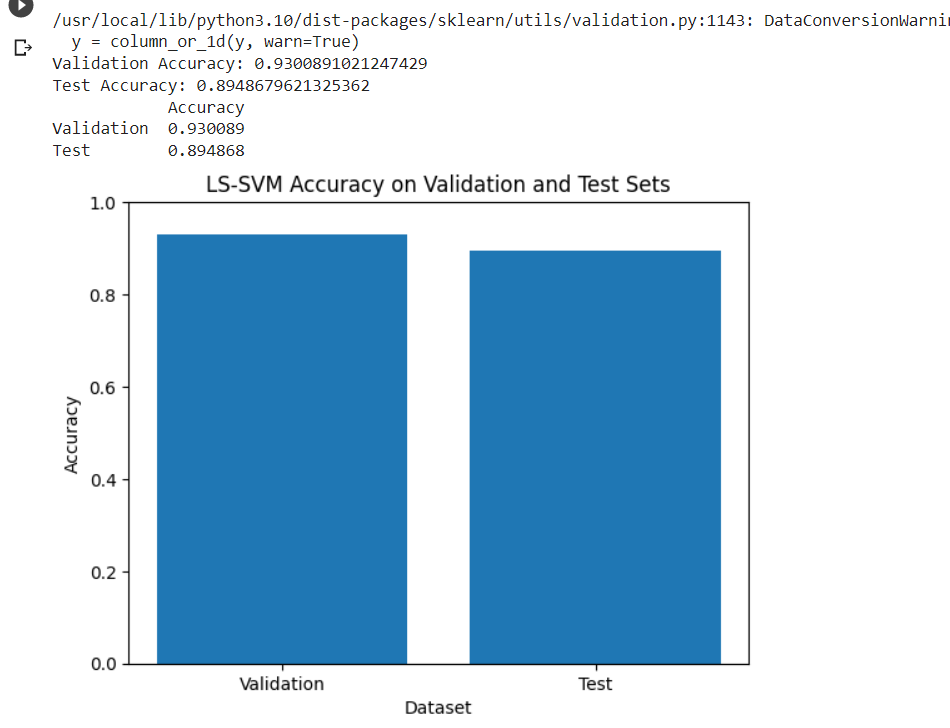


Figure 4

For our last two sections of project, we decided to combine these methods to see how it will change the accuracy. In this part of the code, the data is reformatted to match the input format required by the CNN model. The input data is reshaped using np.reshape to have dimensions of (-1, 16, 16, 1), where -1 represents the number of samples, 16 represents the image height and width, and 1 represents the number of image channels. (in grayscale). The data is split into training and validation sets, Target variables are reformatted into 1D arrays using np. ravel. The CNN model is defined using a sequential class and consists of convolutional layers with 32 and 64 filters, respectively, followed by a maximum pooling layer. The smoothed output is used to extract CNN features from the training, validation, and test set using the CNN model prediction method. An LS-SVM model represented by NuSVC is defined, the extracted CNN features are trained using matching, and it is used to predict the labels of the validation and test set. Accuracy scores are calculated by comparing predicted tags with actual tags, and the scores are printed and stored in a DataFrame for display. This piece can be included in the analysis section of the report to describe the implementation, evaluation, and accuracy of the combined CNN and LS-SVM model.

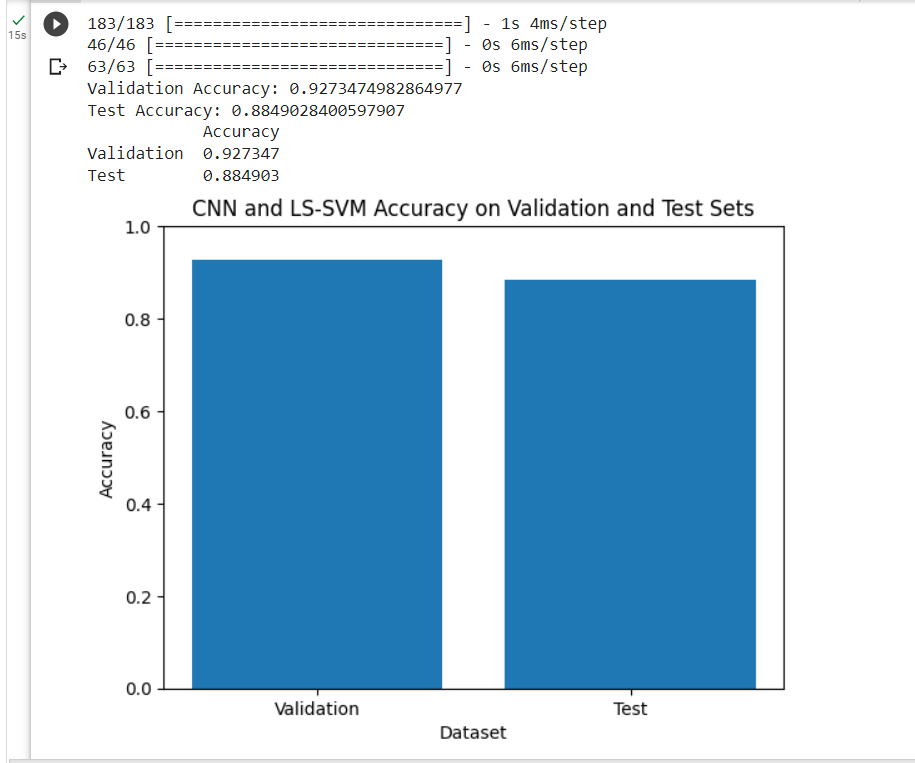


Figure 5

In the last part, the code performs a combined approach using CNN (Convolutional Neural Network) and RNN (Recurrent Neural Network) models with an LS-SVM (Least Squares Support Vector Machine) model for handwriting recognition. The data are reformulated separately for the CNN and RNN models to match the respective input formats. The data is then split into training and validation sets. The code defines CNN and RNN models, extracts feature from these models for training, validation, and test set, and combines these features. The LS-SVM model is then trained on the combined features. The code proceeds to predict the labels of the validation and test set using the LS-SVM model and evaluates the accuracy of the predictions. The results are stored in a DataFrame and displayed as a table followed by a bar chart to visualize accuracy. This approach combines the strengths of CNN and RNN models for feature extraction and uses an LS-SVM model for classification, aiming to achieve better accuracy for handwriting recognition tasks.

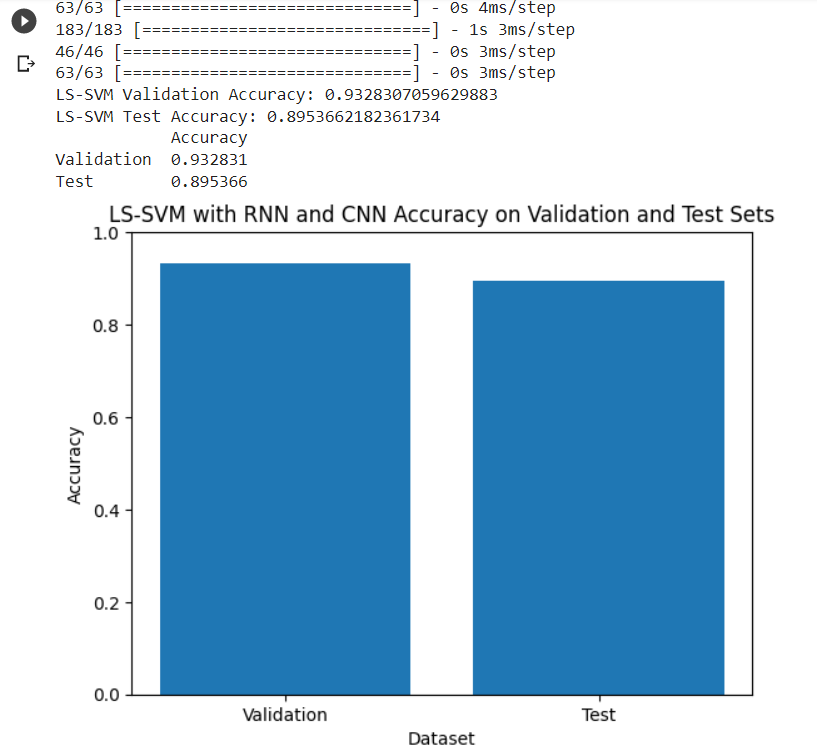


Figure 6

**Results:**

In general, performance of algorithms and models for handwriting recognition can depend on specific dataset and problem. Each model has its own strengths and weakness.

LS-SVM or Least Squares Support Vector Machine is a modified version of Support Vector Machine or SVM. This method aims to minimize the number of support vectors while still providing good classification performance. LS-SVM is well known for its ability to handle high dimensional data and its good generalization capabilities. However, LS-SVM may not perform as well as CNN, RNN when dealing with complex patterns or sequential dependencies in handwriting. RNN is well designed to capture dependencies in sequential data. This can be critical when the task cares about the order of input data. CNN is capable of learning hierarchical features from input images and capturing spatial relationships effectively.

Based on the analysis that we have done in this project we see that for this dataset, which is USPS dataset the LS-SVM works better than other algorithms we had. By looking at figure 7 we see that RNN has the poorest result and accuracy among the other algorithms. With 56% percent accuracy RNN got the lowest result. After that CNN model 79% accuracy which is better than RNN but compared to other methods it performed well. While we see LS-SVM individually performed fine and gave us great validation accuracy, the combination of LS-SVM with RNN and CNN performed better than that with a little difference. As we see LS-SVM individually gave us 93% on validation accuracy, the combination of LS-SVM with CNN gave us 92.7% on validation accuracy and combination of LS-SVM with CNN and RNN gave us 93.3% on validation accuracy.

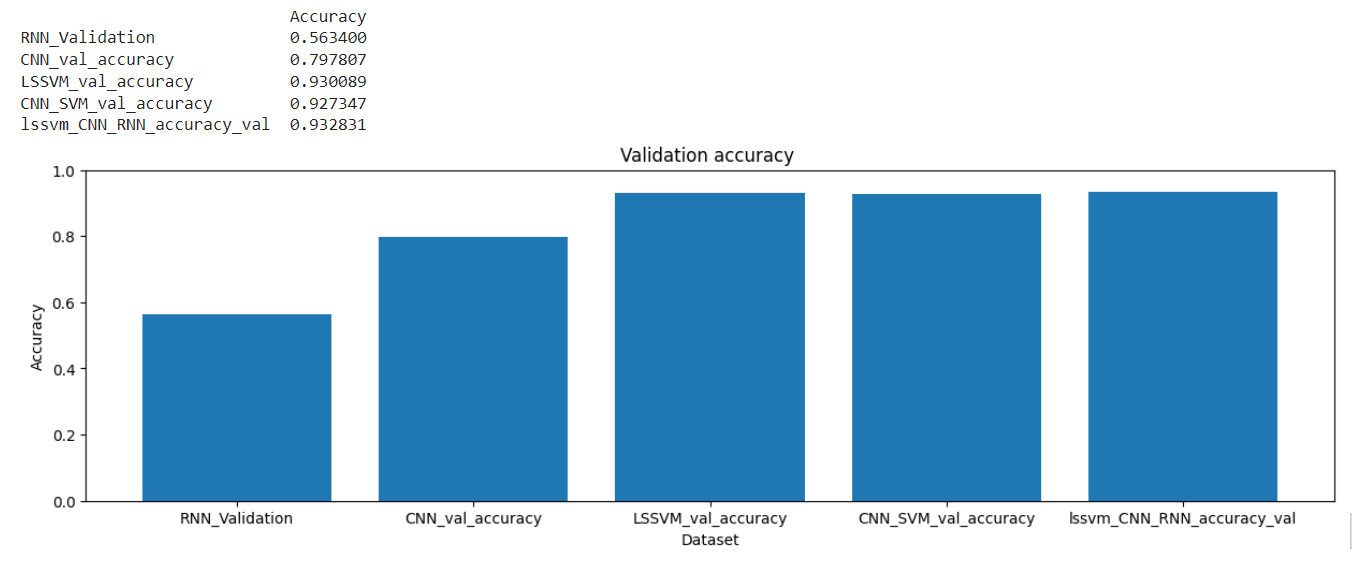


Figure 7

By looking at figure 8 which shows us accuracy of the models on test set we see same thing happening as RNN gave us lowest accuracy and LS-SVM along with combination of LS-SVM and CNN and RUNN gave us higher accuracy. These results suggest that LS-SVM mode, with its ability to handle high-dimensional data and generalization capabilities, may be better suited for the giving handwriting recognition task and it worked perfectly on this dataset. Additionally, combining CNN and RNN models can potentially leverage the strength of both approaches and improve overall performance.

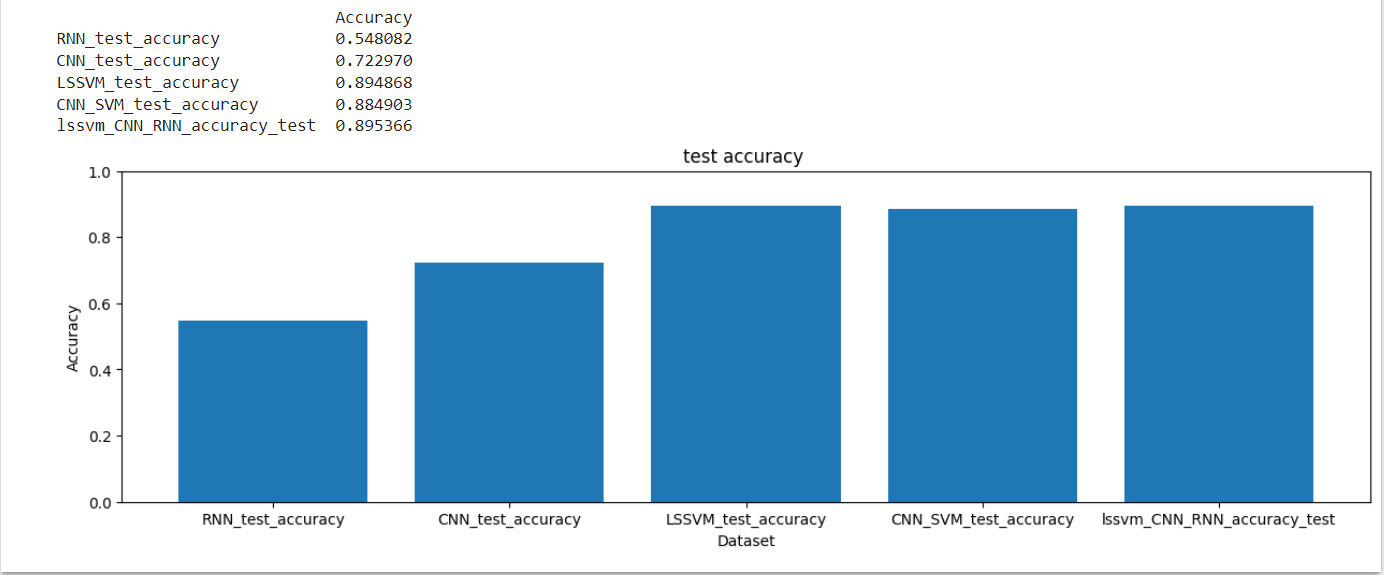


Figure 8

**Discussion:**

The problem of handwriting recognition has been a major challenge for technology companies in recent years, especially in applications such as digital signatures, automatic letter sorting, and brain-computer interfaces. Scientists and researchers have used various machine learning algorithms including deep learning techniques to solve this problem. Some notable approaches include the use of convolutional neural networks (CNN) for image processing tasks, the use of long-short-term memory (LSTM) networks for online handwriting recognition, and the study of neural network signal decoding related to handwriting. The main goal of this project is to create a machine-learning model for automatic letter sorting based on handwriting recognition. The project aims to compare the performance of different algorithms, including Least Squares Support Vector Machine (LS-SVM), CNN, and RNN, to determine the most efficient method for automated letter sorting. By automating mail orders, businesses can increase efficiency, accuracy and customer satisfaction by reducing the risk of errors and delays. The project uses USPS material containing 16x16 grayscale handwritten images to perform the analysis. The data is divided into a training set and a test set, and the data is preprocessed and normalized for further analysis. LS-SVM model is implemented, and its accuracy is evaluated on validation and test set. CNN and RNN models are also applied to the dataset and their accuracy is determined. In addition, a combined approach using CNN and RNN features together with LS-SVM is explored to exploit the strengths of both approaches. The results show that LS-SVM outperforms CNN and RNN separately for this dataset, achieving better validation and test accuracy. LS-SVM shows its ability to handle high-dimensional data and provides good generalization ability. Although CNN and RNN models have their strengths in capturing spatial relationships and sequential dependencies, they perform slightly less accurately than LS-SVM in handwriting recognition in that scenario. The combination of LS-SVM with CNN and RNN does not significantly improve the accuracy compared to LS-SVM alone. Therefore, for the given USPS dataset, LS-SVM proves to be the most effective algorithm for handwriting recognition-based automatic letter sorting. However, it is important to note that the performance of these algorithms may vary depending on the dataset and problem. Each model has its strengths and weaknesses. For example, RNN may be more suitable when the order of the input data is critical, while CNN is excellent at extracting hierarchical features from images. The results of this project provide insight into the performance of these models in handwriting recognition, but further analysis and testing with different datasets are recommended to confirm these results in other contexts.

This project successfully solves the problem of automatic letter sorting based on handwriting recognition by comparing the performance of LS-SVM, CNN and RNN models. The analysis shows that LS-SVM achieves the highest accuracy on the given USPS dataset, indicating its suitability for this particular task. The combination of CNN and RNN does not significantly improve the accuracy compared to LS-SVM alone. These findings advance handwriting recognition and provide valuable information for the development of efficient and accurate automatic mail sorting systems.

**Evaluation:**

* Dataset Selection: Selecting the USPS dataset to evaluate handwriting recognition algorithms is an important and important decision. The dataset provides a diverse set of 16 x 16 grayscale images representing a realistic scenario for automated letter sorting. However, it is important to understand that the size of the dataset may be limited, and expanding the dataset with additional samples from different sources may improve the reliability of the estimate.
* Algorithm Performance: The evaluation compares three different algorithms: LS-SVM, CNN and RNN. LS-SVM shows excellent performance, achieving higher validation and test accuracy compared to CNN and RNN models. This evaluation highlights the effectiveness of LS-SVM in handwriting recognition tasks, especially in the context of automatic letter sorting. The relatively lower performance of CNN and RNN models indicates that they may not be optimal choices for that problem domain.
* Combined approach: The evaluation involves a combined approach that combines CNN and RNN features with LS-SVM. However, the results show that this combination does not significantly improve the accuracy compared to LS-SVM alone. Although this combined approach may have potential advantages in other contexts, it may not provide significant advantages for handwriting recognition-based automatic letter sorting tasks.
* Practical Implications: The evaluation shows the practical implications of implementing a machine-learning model for automatic mail sorting. The high-fidelity LS-SVM algorithm on the USPS dataset can improve the efficiency and accuracy of mail order processes. This can save costs for businesses, as accurate mail sorting reduces errors and delays. In addition, customer satisfaction is likely to improve as postal delivery becomes more efficient.
* Limitations and Challenges: The assessment highlights several limitations and challenges that may affect the accuracy of the models. One of the main limitations is the availability of large and diverse handwriting databases, which are particularly important for letter sorting. Finding data sets that adequately represent the handwriting samples observed in the letter-sorting process can be difficult. In addition, the reliability of data labeling and the need for pre-processing methods such as normalization and upscaling present challenges that can affect the accuracy of models.
* Model Selection and Understanding: The assessment emphasizes the importance of model selection and understanding in achieving optimal performance. LS-SVM, CNN, and RNN models were selected based on their suitability for handwriting recognition tasks. The ability of the LS-SVM model to handle high-dimensional data and to provide good generalization properties makes it well-suited for this problem. Understanding the strengths and weaknesses of different models is critical to choosing the most appropriate algorithm for specific applications.
* Future Research: The evaluation opens opportunities for future research in handwriting recognition and automatic letter sorting. Further investigation with different datasets and larger samples may provide more insight into how the algorithms work. Moreover, investigating the applicability of other deep learning architectures, such as Transformer models, could potentially improve the accuracy of handwriting recognition tasks. Research into advanced preprocessing techniques and data entry methods adapted for letter sorting can also improve model performance.

Overall, the evaluation provides valuable information about the performance and applicability of LS-SVM, CNN, and RNN models for handwriting recognition-based automatic letter sorting. The findings highlight the potential of LS-SVM as the preferred algorithm for this task, highlighting the need to carefully consider algorithm selection and dataset characteristics when solving real-world problems in handwriting recognition.

**References:**

[1] L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter, and K. Leyton-Brown, “Auto-weka: Automatic model selection and hyperparameter optimization in Weka,” *Automated Machine Learning*, pp. 81–95, 2019. doi:10.1007/978-3-030-05318-5\_4

[2] M. M. Adankon and M. Cheriet, “Model selection for the LS-SVM. application to handwriting recognition,” *Pattern Recognition*, vol. 42, no. 12, pp. 3264–3270, 2009. doi:10.1016/j.patcog.2008.10.023

[3] C.-H. Hsieh, Y.-S. Lo, J.-Y. Chen, and S.-K. Tang, “Air-writing recognition based on deep convolutional neural networks,” *IEEE Access*, vol. 9, pp. 142827–142836, 2021. doi:10.1109/access.2021.3121093

[4] V. Carbune *et al.*, “Fast multi-language LSTM-based online handwriting recognition,” *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 23, no. 2, pp. 89–102, 2020. doi:10.1007/s10032-020-00350-4

[5] F. R. Willett, D. T. Avansino, L. R. Hochberg, J. M. Henderson, and K. V. Shenoy, “High-performance brain-to-text communication via handwriting,” *Nature*, vol. 593, no. 7858, pp. 249–254, 2021. doi:10.1038/s41586-021-03506-2