

Literature Review

Adrian Manhey

amanhey@oxy.edu

Occidental College

1 Problem Context

Convolutional Neural Networks (CNNs) has been widely successful in applications for classifying images but the application of these models to tabular data has not been as widespread. The main advantage of using CNNs is that feature detection is automatic and can be done without supervision. Additionally, through unique layer designs, like covariance pooling, CNNs are able to extract higher-order statistics and nonlinear correlations [6] and the average computation efficiency of convolutional layers is 3.1 times that of fully connected layers, since convolutional layers already have high GPU resource utilization [5].

In image classification, the internal data structure of a picture is detected by the neural network layers. Some of these structures, like colors and shapes, can be detected by human sense but we are unable to detect such structures of tabular data. Using CNNs on tabular data, by first converting them into images, may give insight into some internal structure that can be visualized as well.

Furthermore, some of the CNNs implemented are able to perform as well as non-Neural Network models, even when not optimized. This approach may provide an alternative path to achieve or surpass non-Neural Network models with the right optimization to the implemented model's parameters.

2 Prior Work

Still a novel approach, some studies of CNNs for tabular data have been conducted in biomedicine and finance, where tabular data is prevalent. Of the related studies, 6 out of 10 of them were in the biomedical field, many using gene expression data. Of the 4 others, 3 involved time-series datasets. In 2015, Wang et al. (2015) [7], seemingly the first to apply CNNs to tabular data, encoded their data in Gramian Angular Summation/Difference Fields (GASF/GADF) and Markov Transition Fields (MTF) and then applied a tile CNN to learn high-level features and reduce the imputation Mean Squared Error on their test data by 12.18

Since then various implementations have been used to create images suitable for classification. Basu et al. [1] converted financial time-series data of stocks into images

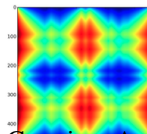


Figure 1. Wang 2015 Gramian Angular Summation Field



Figure 2. Basu 2022 Financial Peak with Decline

by considering only certain features. They used the High, Low, and Closing price of S&P 500 index stocks, in a range of an odd-squared number of days (e.g. 9, 25, 49, etc.), to create RGB values of pixels. Afterwards, they used a pre-trained Siamese Type Neural Network to classify their images, which can be seen in Figure 2.

Zhu et al. (2021) develop Image Generator for Tabular Data (IGTD), whose conversion process consisted of assigning features to pixel positions so similarity of features can be expressed by distance of pixels [8]. The algorithm searches for an optimized assignment by minimizing the difference between the ranking of distances between features and the ranking of distances between their assigned pixels in the image.

Bazgir et al. (2020) developed REFINED (REpresentation of Features as Images with NEighborhood Dependencies) [2], which uses the Bayesian multidimensional scaling to create a map displaying the relative positions of the features, given a table of the distances between them. The features are then assigned to image pixels according to the mapping and a hill climbing algorithm is applied to locally optimize the arrangement of feature positions in the image.

Ma et al. (2018) developed OmicsMapNet to convert gene expression data of cancer patients into 2-D images for the prediction of tumor grade using CNNs [4]. OmicsMapNet utilizes annotated genes to construct images via TreeMap, so that genes with similar molecular functions are closely located in the image, in a similar fashion to the previous two implementations.

Lastly, the method implemented in Buturović et al. (2020) had similarities to some of the implementations in

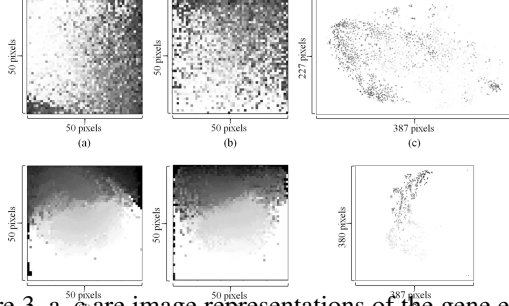


Figure 3. a–c are image representations of the gene expression profile of the SNU-61 cell line generated by IGTD, REFINED, and DeepInsight, respectively. d–f are image representations of molecular descriptors of Nintedanib, generated by IGTD, REFINED, and DeepInsight, respectively.[8]

terms of data or method. Their goal was to classify bacterial and viral infections from blood samples [3] and similarly to Basu et al. (2022), they used sequences of odd-square length but converted their data into a kernel. A kernel, also known as convolution matrix or mask, is a small matrix used to create a filter on an image. This happens by taking the convolution matrix, usually a 3x3 or 5x5 matrix, and iteratively convoluting it with every 3x3 or 5x5 pixel segment of a base image, such as a rose.

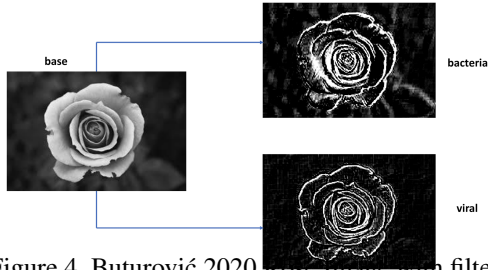


Figure 4. Buturović 2020 rose image with filters

The convolution process will be discussed more in the section 3 of this review. The resulting filtered image (Figure 4) is used with a default ResNet model to classify them. This study showed that unoptimized CNN can achieve the same level of accuracy as optimized CNNs. Yet it also left some questions unanswered such as the impact of different base images, if a set of base images could be used instead of just one, and especially, what kind of results could be achieved by an optimized CNN. These questions are part of the main considerations for this project.

3 Technical Background

This implementation will require knowledge of both CNN and non-CNN models in order to compare the efficacy of each. Buturović et al. (2020) implemented 6 different non-CNN models for comparison with ResNet: XGBoost,

LightGBM, SVM with linear and RBF kernel, multi-layer perceptron (MLP) and regularized logistic regression. In terms of comparison, at least one or two optimized non-CNN algorithms will be implemented. Which ones will be chosen after further investigation of the non-CNN models implemented in other studies and answering questions about their strengths, weaknesses, and in what areas may we want to choose one over another? However, Ma et al. (2018), Bazgir et al. (2020), and Zhu et al. (2021) each seem to have their implementations available to the public, so comparisons with other tabular conversion methods also exists as an avenue.

It is likely that the focus of this project will use the kernel method for conversion. Kernels work by creating a copy of an image with a filter applied using the underlying matrix structure of images. This may be a little difficult to understand at first so let's consider an example.

Let A be a sample matrix of pixels, C be the convolution matrix, and R be our resulting convoluted matrix. Let each be a $n \times n$ matrix where $n = 3$. Our convolution can be represented by $A \cdot C = R$. Consider the following example where $A \cdot C$ is given by,

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \cdot \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

The kernel used here is the one that approximates the Gaussian blur filter of an image. The convolution of A with respect to C can be computed by

$$\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (A_{ij} * C_{ij}) = 1(2) + 2(2) + 3(1) + 4(2) + 5(4) + 6(2) + 7(1) + 8(2) + 9(1)$$

This sum is equal to 81 and R will become the following:

$$= \frac{1}{16} \begin{bmatrix} -- & -- & -- \\ -- & 81 & -- \\ -- & -- & -- \end{bmatrix}$$

The blank (--) spaces representing the empty spaces. These spaces will be filled as the convolution matrix iterates through the different pixel sub-matrices of an image.

Using a method such as this, this project aims to develop off Buturović 2020's research. A main development will be to implement ResNet on a new dataset, test the performance on metrics such as accuracy and time, and then see the difference in performance when the model when optimized. The optimization process will need to be planned out and different methods will need to be weighed against each other. However, Dr. Basu, of Basu et al. 2022, has

suggested using a random forest algorithm with the CNN as part of the optimization process. A random forest tree creates multiple decision trees and searches for the best feature among a random subset of features, which increases the diversity of the results. Finally, in relation to kernels, it would be desirable to understand the different kinds of convolution matrices and the effect that each one has on an image as well as if some are more effective than others at classification.

References

- [1] Treena Basu et al. “A Novel Implementation of Siamese Type Neural Networks in Predicting Rare Fluctuations in Financial Time Series”. In: *Risks* 10.2 (2022). ISSN: 2227-9091. DOI: 10 . 3390 / risks10020039. URL: <https://www.mdpi.com/2227-9091/10/2/39>.
- [2] Omid Bazgir et al. “Representation of features as images with neighborhood dependencies for compatibility with convolutional neural networks”. In: *Nature Communications* 11.1 (Sept. 2020), p. 4391. DOI: 10 . 1038 / s41467-020-18197-y. URL: <https://doi.org/10.1038/s41467-020-18197-y>.
- [3] Ljubomir Buturović and Dejan Miljković. “A novel method for classification of tabular data using convolutional neural networks”. In: *bioRxiv* (2020). DOI: 10 . 1101 / 2020 . 05 . 02 . 074203. eprint: <https://www.biorxiv.org/content/early/2020/05/03/2020.05.02.074203.full.pdf>. URL: <https://www.biorxiv.org/content/early/2020/05/03/2020.05.02.074203>.
- [4] Shiyong Ma and Zhen Zhang. “OmicsMapNet: Transforming omics data to take advantage of Deep Convolutional Neural Network for discovery”. In: *ArXiv abs/1804.05283* (2018).
- [5] T.M. Navamani. *Chapter 7 - Efficient Deep Learning Approaches for Health Informatics*. Ed. by Arun Kumar Sangaiah. Academic Press, 2019, pp. 123–137. ISBN: 978-0-12-816718-2. DOI: <https://doi.org/10.1016/B978-0-12-816718-2.00014-2>. URL: <https://www.sciencedirect.com/science/article/pii/B9780128167182000142>.
- [6] Qilong Wang et al. “Deep CNNs Meet Global Covariance Pooling: Better Representation and Generalization”. In: *CoRR abs/1904.06836* (2019). arXiv: 1904.06836. URL: <http://arxiv.org/abs/1904.06836>.
- [7] Zhiguang Wang and Tim Oates. “Imaging Time-Series to Improve Classification and Imputation”. In: *ArXiv abs/1506.00327* (2015).
- [8] Yitan Zhu et al. “Converting tabular data into images for deep learning with convolutional neural networks”. In: *Scientific Reports* 11.1 (May 2021), p. 11325. DOI: 10 . 1038 / s41598-021-90923-y. URL: <https://doi.org/10.1038/s41598-021-90923-y>.