Deep Learning Pipeline for Classification of Time Series Data

Ryan J. Urbanowicz, Arjun Shah

Background

- Time Series data is a collection of observed measurements that maintain a sequential order
  - Ex: Audio, EEG, Text, etc.
- A major problem regarding time series data is time series classification (TSC). TSC seeks to predict a certain class of a new time series based on models training data.
- With regards to this PURM project, we specifically sought to classify EEG data which can be used to identify certain micro sleep events
- In the current literature and online repositories, there are no all-encompassing pipelines that contain newly developed methods of TSC analysis which is why we decided to develop one
- Specific attributes of time series data
  - Seasonality: Variations that repeats itself over time
  - Trend: Gradual upward or downward movements
  - Level: Average value or moving average value of the series
  - Stationarity: Time series whose mean and variance are constant over time
- Deep learning takes in input data and extracts features automatically in order to classify the data set, unlike regular machine learning where feature extraction is manually completed. Recently, convolutional neural networks are being utilized in order to analyze TS data
- **GOAL:** Develop an all-encompassing pipeline (AutoMLPipe-TSC) that preprocesses univariate and multivariate time series data in order to apply recent contributions in computer vision technology such as YOLO and SSD to classify said data.

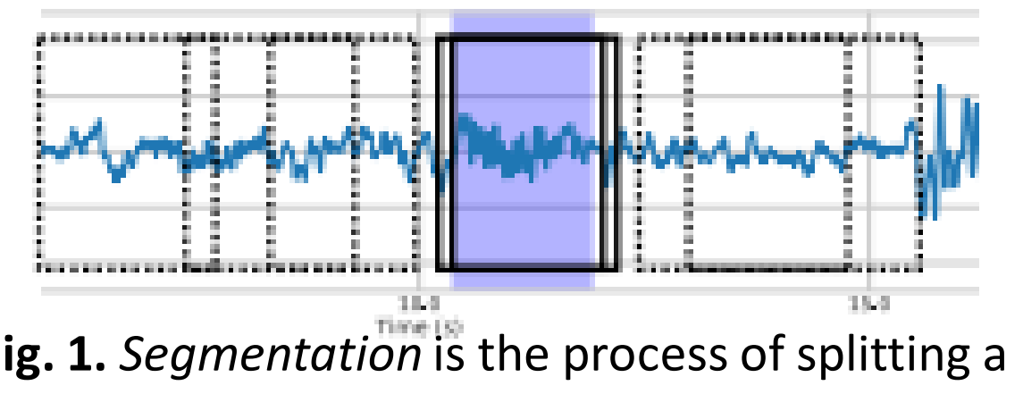


Fig. 1. *Segmentation* is the process of splitting a time series into discrete intervals to reveal underlying properties of the data. This example shows the segmentation of EEG data in sleep event classification

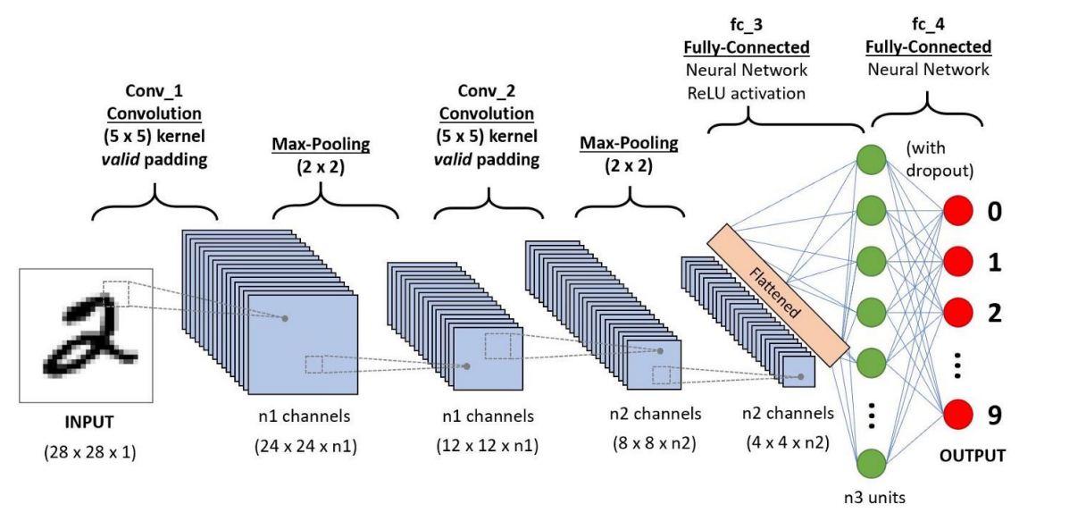


Fig. 2. A CNN is a deep learning method that extracts features through multiple convolutions and pooling layers. During classification, the data is flattened, and a loss function is applied to optimize the model

Current Research/Pipelines

- **Sktime:** Sktime is a unified framework for time series analysis in classification and forecasting. This was a major framework utilized in the pipeline since other advanced models were based on it.
- **TapNet**: TapNet is a multivariate time series classification tool that can deal with high amounts of data as an attentional prototype network.
- **InceptionTime:** Extremely effective at handling large time series data sets since it utilizes the InceptionTime CNN which outperforms the traditional HIVE-COTE method.
- **MINIROCKET:** MR is a fast deterministic transform for TSC by utilizing random CNNs to train a linear classifier.
- **Time Warping Invariant Echo State Networks:** Echo State Networks are an alternative to CNNs in time series analysis but have usually been too computationally complex. However this research shows how ESNs are more accurate than traditional manual feature learning methods.

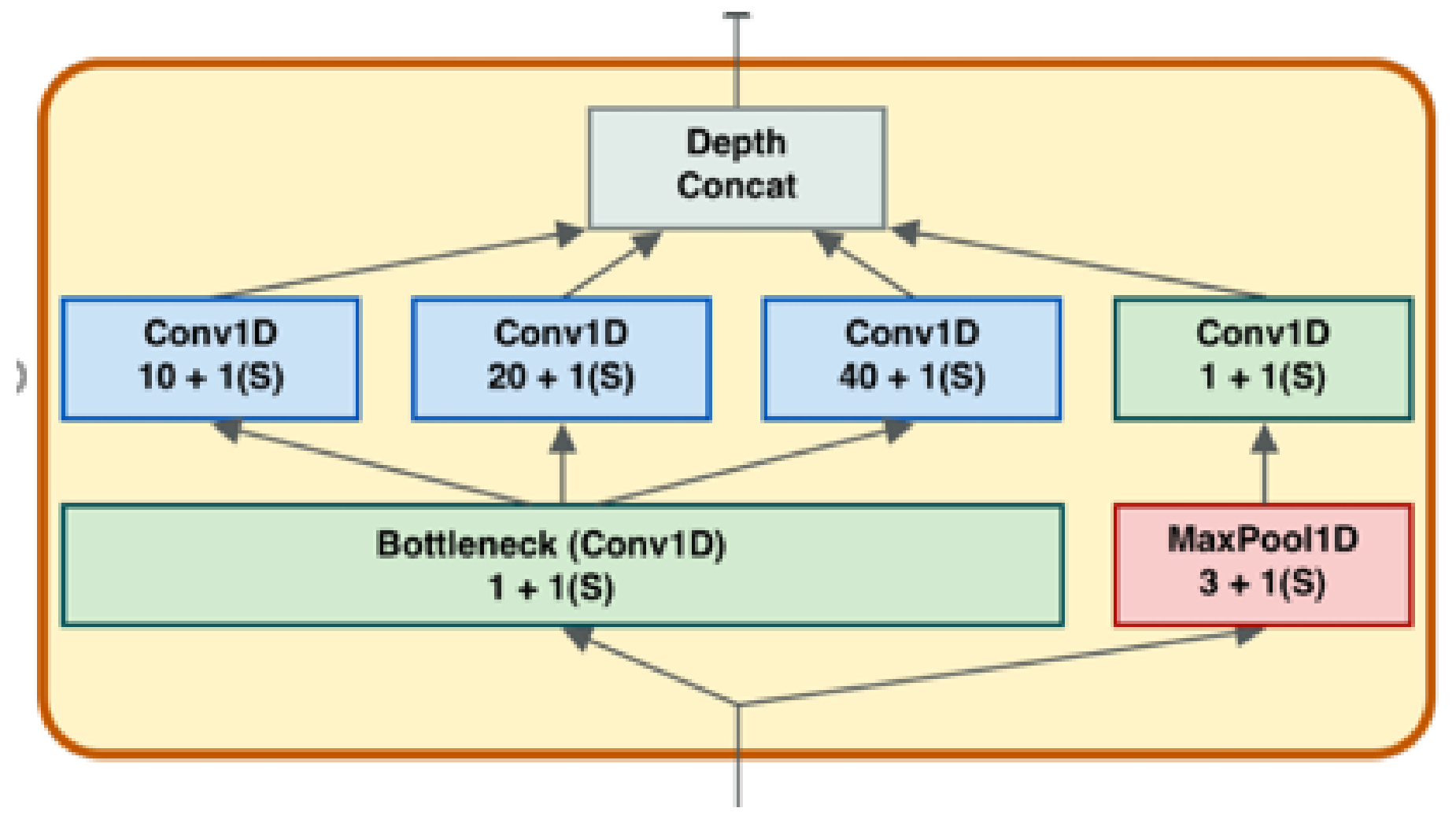


Fig. 6. This is the inception module for inception time. The bottleneck layer reduces the dimensionality of the inputs which reduces the training time. The input is then sent through 3 CNNs and one pooled CNN where it is concatenated together.

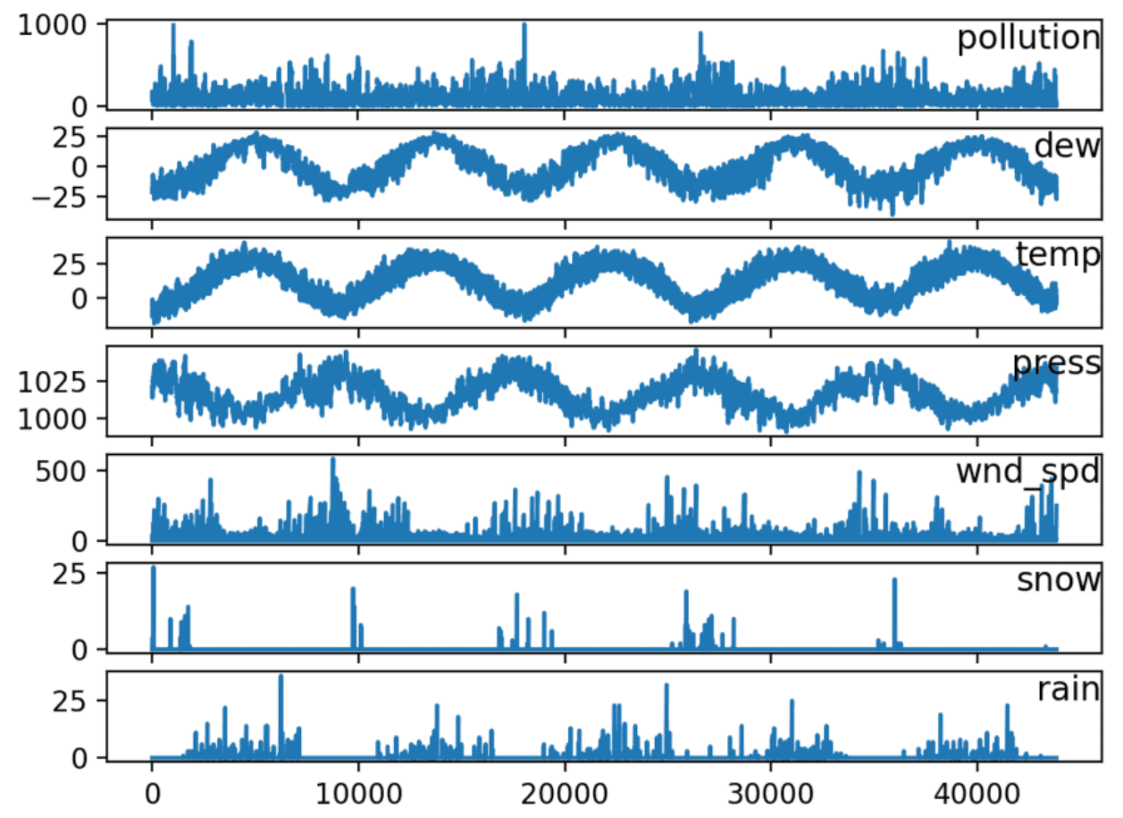


Fig. 7. Here is an example of a multivariate time series data set that can be analyzed by TapNet. This data set represents environmental conditions that can be used to classify pollution levels.

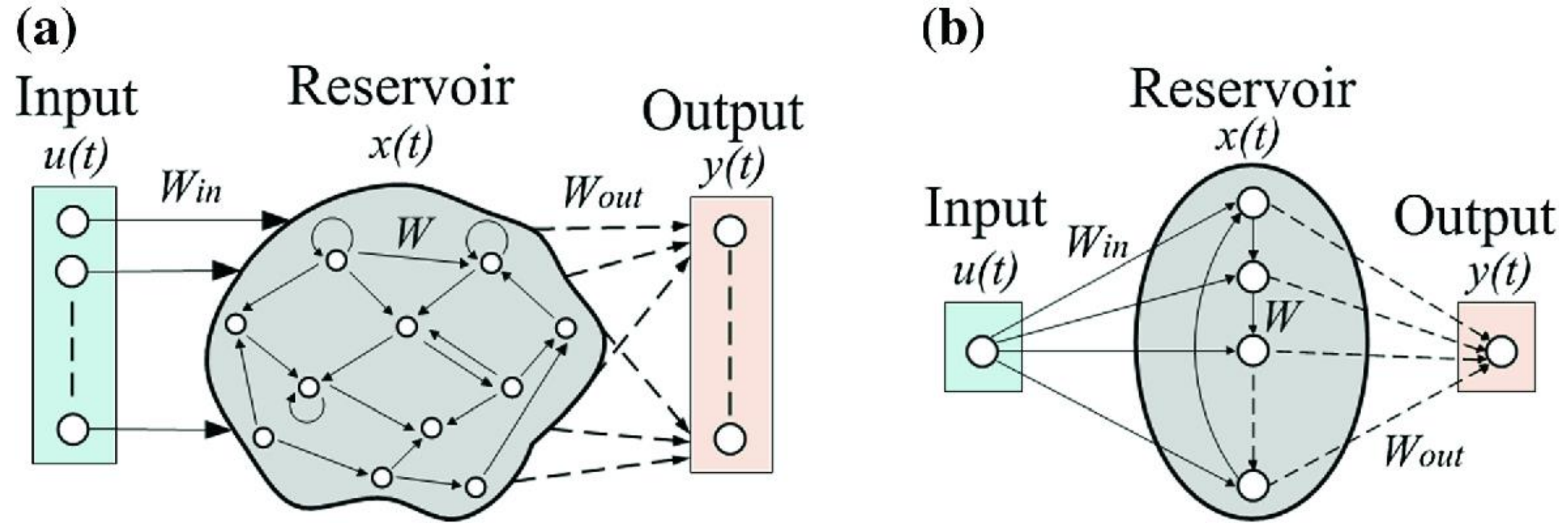


Fig. 8. Part (a) represents an ESN and how it utilizes a connected reservoir to train the data set compared to a normal CNN in part (b)

Strategies for Analyzing Time Series Data

- Segmentation: By segmenting data, we make can analyze hidden properties of the underlying data. A well known example of this is sound data, where the time series can be segmented by different voices or sounds. Through segmentation, our training time drastically decreases as the model is only searching for local features in each discrete window instead of both local and global features.
  - Types of segmentation:
    - Sliding Window: Set of default events are created and are a specific discrete length
    - Rolling Window: A window of a set value moves one increment along the time series (more time required than sliding window)
    - Adjusted Sliding Window: Like the normal sliding window, the adjusted sliding window finds the center of the default events and adjusts the bounds of the window
- Image Classification Techniques: Since computer vision and image classification techniques have been around longer and studied more than regular time series techniques, it is useful to change the one dimensional time data to a two-dimensional image that can be analyzed.
  - Gramian Angular Fields: A GAF essentially scales the data, converts it to polar, and then constructs a gram matrix using the inner product. The density of the gram matrix is plotted which can be analyzed using image classification techniques
  - Markov Transition Fields: More complicated than a GAF, a MTF transforms 1D time series data into a Markov transition matrix by assigning a probability along the temporal order
- Convolutional Neural Networks: CNNs extract high level features by applying multiple filters in convolutional and pooling layers to capture spatial and temporal dependencies in images.
  - You Only Look Once (YOLO): Applies a single neural network (NN) to an image, divides the image into regions and then predicts probabilities of bounding boxes. Each bounding box represents an area with an associated probability of an object being in said box
  - Single Shot MultiBox Detector (SSD): A grid is placed over the image and each individual cell is responsible for classifying If an object is in it or not. Then an anchor box bounds the cells that contain the object.
  - Tiled CNN: A CNN that uses a tiled pattern of weights where neighboring units are pooled over to classify objects in the image.

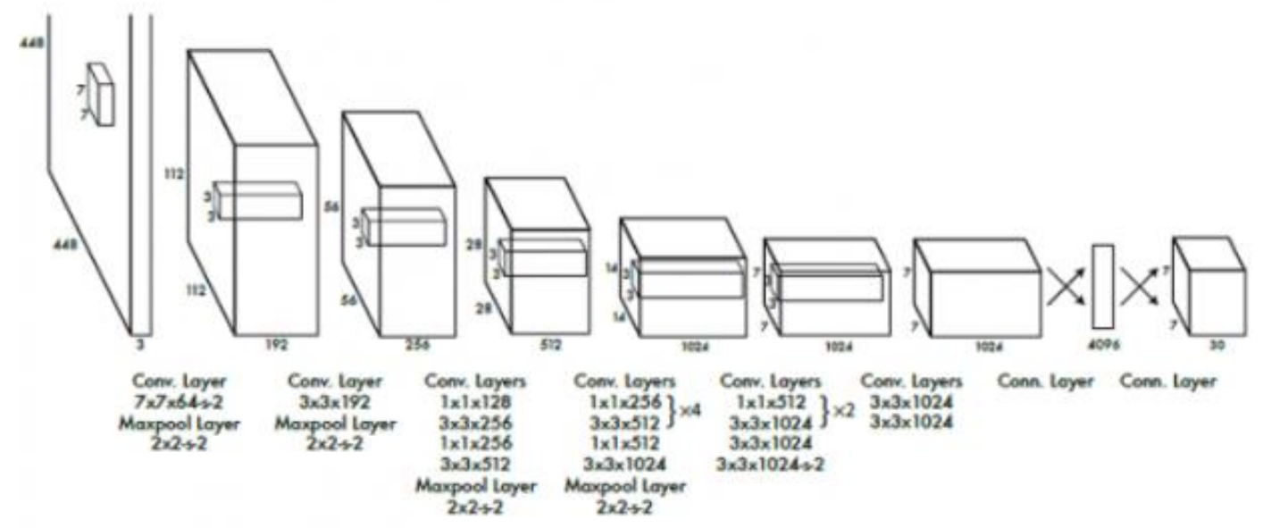


Fig. 3. YOLO model example. YOLO is extremely computationally efficient since the model only runs once. The only struggle with YOLO is the difficulty in detecting small objects/events, unlike SSD, which is slower but more accurate.

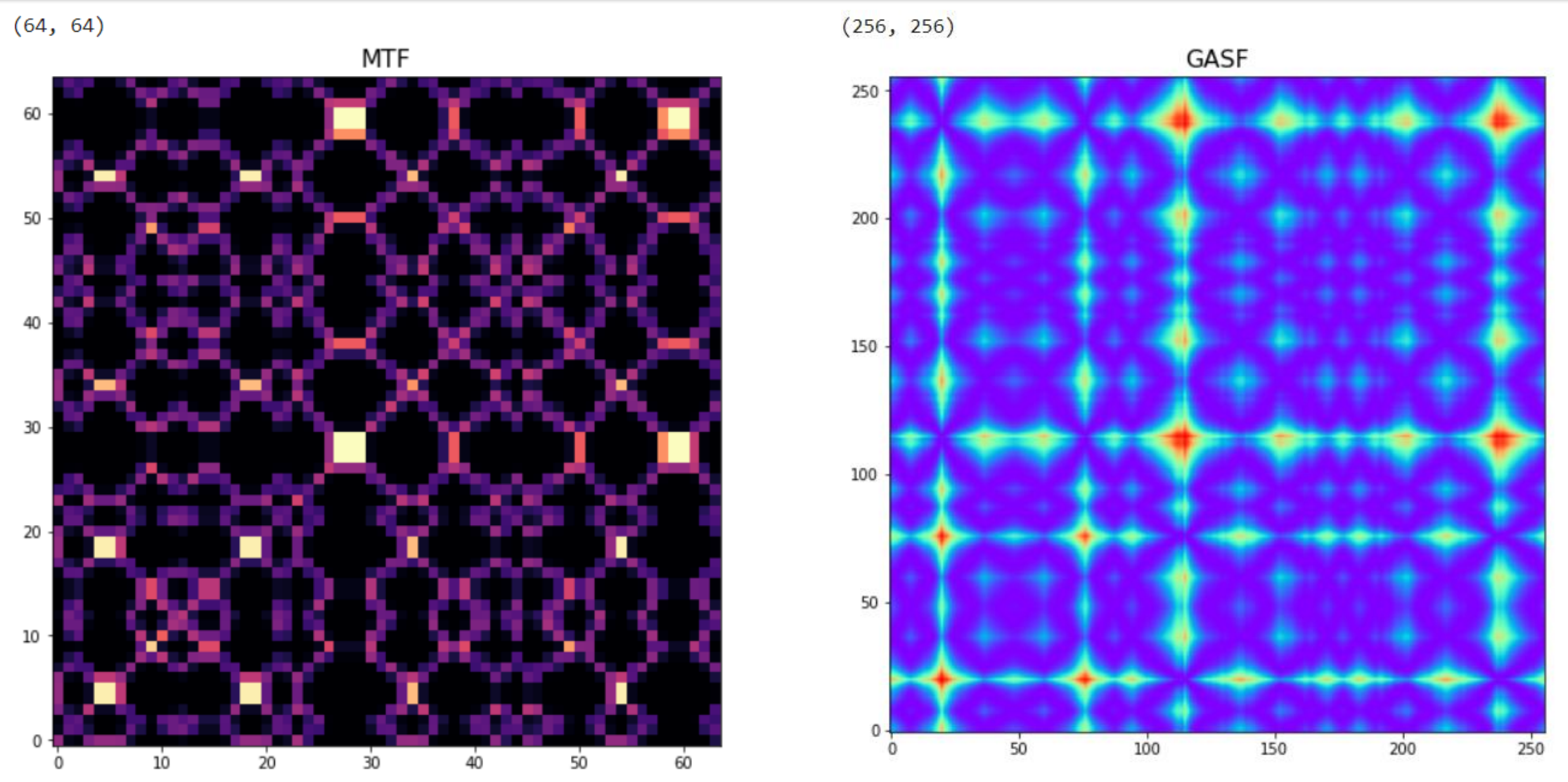


Fig. 4. These are examples of a Markov Transition Field (left) and a Gramian Angular Field (right). Both are a two dimensional image that can be used in regular image classification models.

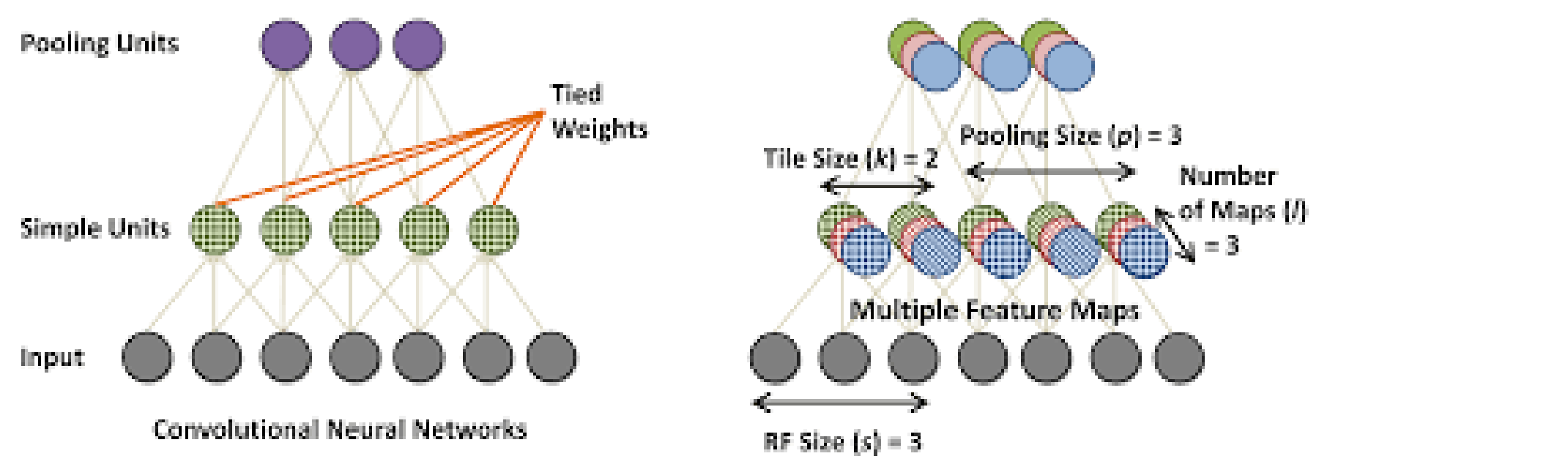


Fig. 5. In a tiled CNN, weights are given to each tile to maintain the spatial data of the image. Similar to regular CNNs, tiled CNNs have a small number of learned parameters for increased scalability

Analysis & Preliminary Results

- AutoMLPipe-TSC is an extension of the already built AutoMLPipe-BC which handles general binary classification
- To build the basis of the pipeline, I utilized the Sktime unified framework since it already has prebuilt classification techniques that can be applied to time series sets
- HIVE-COTE was used as the baseline algorithm all future algorithms would be compared to since it is the oldest/most popular TSC algorithm
- Besides YOLO, I have started the implementation of InceptionTime into AutoMLPipe-TSC because of its general breadth when handling different time series sets
- This pipeline is a combination of all the current cutting-edge method in time series classification
- The UCR data set was used for testing since it is a collection of multiple time series data sets
- The data sets used for training were all presegmented which allowed for an easier analysis process

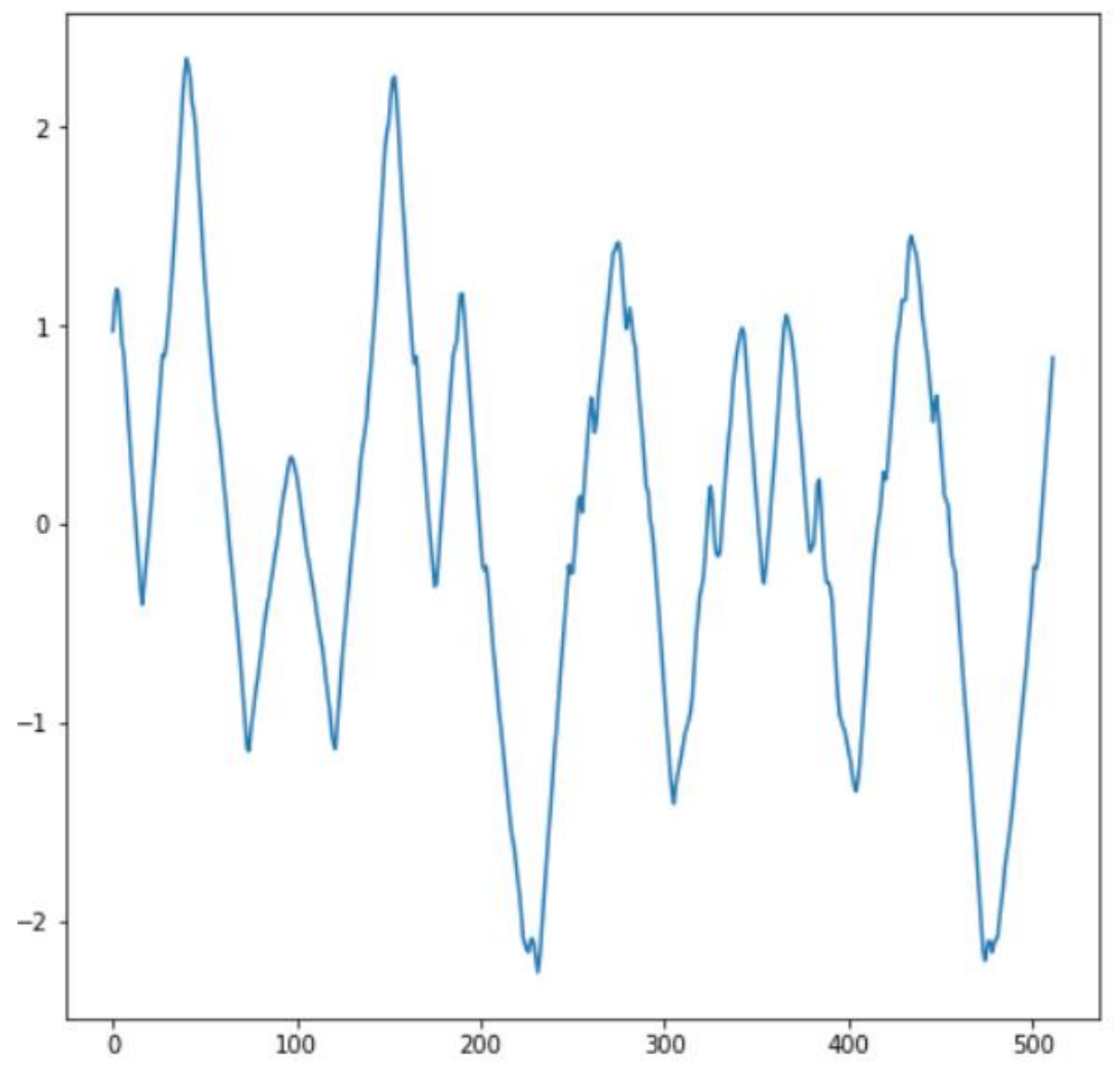
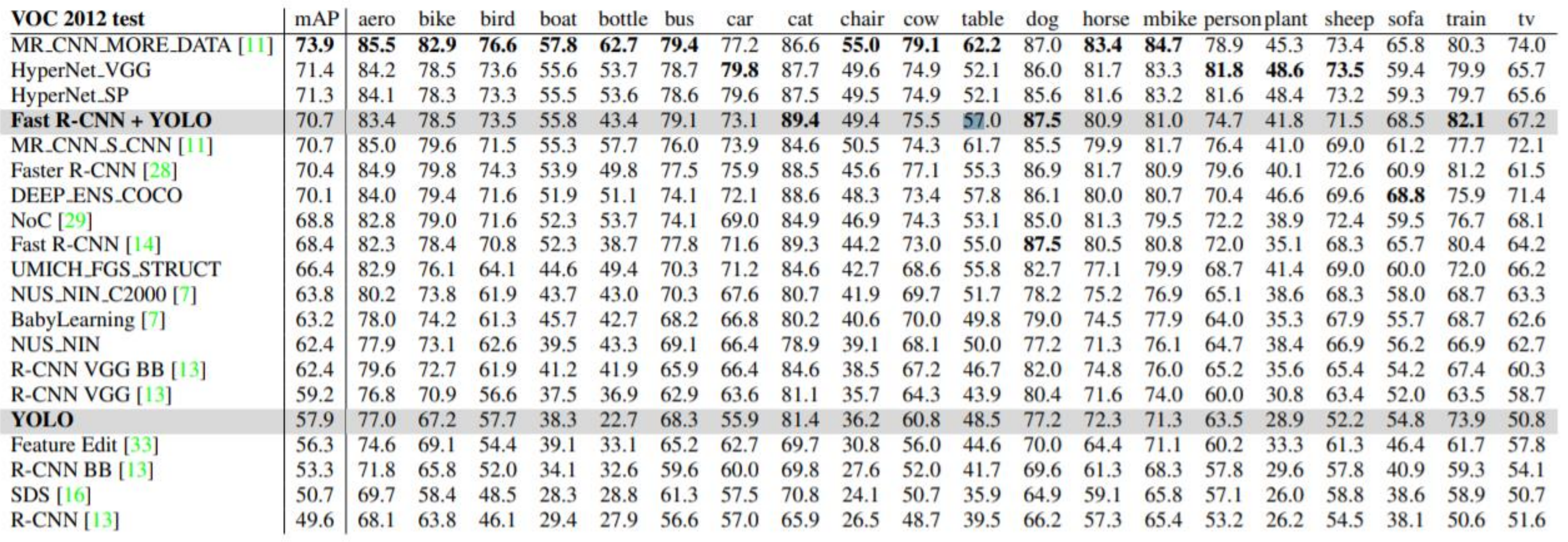


Fig. 9. Example of the pipeline preprocessing a time series data. The data has been cleaned (missing values replaced) and normalized.



VOC 2012 test	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
MR_CNN_MORE_DATA [11]	**73.9**	**85.5**	**82.9**	**76.6**	**57.8**	**62.7**	**79.4**	77.2	86.6	**55.0**	**79.1**	**62.2**	87.0	**83.4**	**84.7**	78.9	45.3	73.4	65.8	80.3	74.0
HyperNet_VGG	71.4	84.2	78.5	73.6	55.6	53.7	78.7	**79.8**	87.7	49.6	74.9	52.1	86.0	81.7	83.3	**81.8**	**48.6**	**73.5**	59.4	79.9	65.7
HyperNet_SP	71.3	84.1	78.3	73.3	55.5	53.6	78.6	79.6	87.5	49.5	74.9	52.1	85.6	81.6	83.2	81.6	48.4	73.2	59.3	79.7	65.6
Fast R-CNN + YOLO	70.7	83.4	78.5	73.5	55.8	43.4	79.1	73.1	**89.4**	49.4	75.5	57.0	**87.5**	80.9	81.0	74.7	41.8	71.5	68.5	**82.1**	67.2
MR_CNN_S_CNN [11]	70.7	85.0	79.6	71.5	55.3	57.7	76.0	73.9	84.6	50.5	74.3	61.7	85.5	79.9	81.7	76.4	41.0	69.0	61.2	77.7	72.1
Faster R-CNN [28]	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
DEEP_ENS_COCO	70.1	84.0	79.4	71.6	51.9	51.1	74.1	72.1	88.6	48.3	73.4	57.8	86.1	80.0	80.7	70.4	46.6	69.6	**68.8**	75.9	71.4
NoC [29]	68.8	82.8	79.0	71.6	52.3	53.7	74.1	69.0	84.9	46.9	74.3	53.1	85.0	81.3	79.5	72.2	38.9	72.4	59.5	76.7	68.1
Fast R-CNN [14]	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	**87.5**	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
UMICH_FGS_STRUCT	66.4	82.9	76.1	64.1	44.6	49.4	70.3	71.2	84.6	42.7	68.6	55.8	82.7	77.1	79.9	68.7	41.4	69.0	60.0	72.0	66.2
NUS_NIN_C2000 [7]	63.8	80.2	73.8	61.9	43.7	43.0	70.3	67.6	80.7	41.9	69.7	51.7	78.2	75.2	76.9	65.1	38.6	68.3	58.0	68.7	63.3
BabyLearning [7]	63.2	78.0	74.2	61.3	45.7	42.7	68.2	66.8	80.2	40.6	70.0	49.8	79.0	74.5	77.9	64.0	35.3	67.9	55.7	68.7	62.6
NUS_NIN	62.4	77.9	73.1	62.6	39.5	43.3	69.1	66.4	78.9	39.1	68.1	50.0	77.2	71.3	76.1	64.7	38.4	66.9	56.2	66.9	62.7
R-CNN VGG BB [13]	62.4	79.6	72.7	61.9	41.2	41.9	65.9	66.4	84.6	38.5	67.2	46.7	82.0	74.8	76.0	65.2	35.6	65.4	54.2	67.4	60.3
R-CNN VGG [13]	59.2	76.8	70.9	56.6	37.5	36.9	62.9	63.6	81.1	35.7	64.3	43.9	80.4	71.6	74.0	60.0	30.8	63.4	52.0	63.5	58.7
YOLO	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
Feature Edit [33]	56.3	74.6	69.1	54.4	39.1	33.1	65.2	62.7	69.7	30.8	56.0	44.6	70.0	64.4	71.1	60.2	33.3	61.3	46.4	61.7	57.8
R-CNN BB [13]	53.3	71.8	65.8	52.0	34.1	32.6	59.6	60.0	69.8	27.6	52.0	41.7	69.6	61.3	68.3	57.8	29.6	57.8	40.9	59.3	54.1
SDS [16]	50.7	69.7	58.4	48.5	28.3	28.8	61.3	57.5	70.8	24.1	50.7	35.9	64.9	59.1	65.8	57.1	26.0	58.8	38.6	58.9	50.7
R-CNN [13]	49.6	68.1	63.8	46.1	29.4	27.9	56.6	57.0	65.9	26.5	48.7	39.5	66.2	57.3	65.4	53.2	26.2	54.5	38.1	50.6	51.6

Fig. 10. Comparison of YOLO to other image classification techniques and CNNs on the VOC 2012 dataset (Redmon 2016)

Conclusions

Conclusion

- Based on the preliminary data set results, the pipeline is able to handle unsegmented data and apply image classification techniques
- Future work would include a method to manually segment the data set using methods such as rolling window or adjusted sliding window

FUTURE WORK

- Develop an addition to the pipeline that handles vision transformers, a new method in image classification which is faster than most traditional CNNs
- Regarding Penn Medicine, AutoMLPipe-TSC can be applied to handle medical data such as EEG data
- For example, it would be interesting to explore how the pipeline classifies sleep apnea data

References

1) H C, D. (2020). An Overview of You Only Look Once: Unified, Real-Time Object Detection. *International Journal for Research in Applied Science and Engineering Technology*, *8*(6), 607–609. https://doi.org/10.22214/ijraset.2020.6098

2) University of Pennsylvania, Urbanowicz, R., & Suri, P. (2020, September). *A RIGOROUS MACHINE LEARNING ANALYSIS PIPELINE FOR BIOMEDICAL BINARY CLASSIFICATION: APPLICATION IN PANCREATIC CANCER NESTED CASE-CONTROL STUDIES WITH IMPLICATIONS FOR BIAS ASSESSMENTS*. arXiv.

3) S. Chambon, V. Thorey, P.J. Arnal, E. Mignot, A. Gramfort,
DOSED: A deep learning approach to detect multiple sleep micro-events in EEG signal, Journal of Neuroscience Methods, Volume 321, 2019,Pages 64-78,ISSN 0165-0270, https://doi.org/10.1016/j.jneumeth.2019.03.017.

4) Jin, Linpeng & Dong, Jun. (2016). Ensemble Deep Learning for Biomedical Time Series Classification. Computational Intelligence and Neuroscience. 2016. 1-13. 10.1155/2016/6212684.

5) Wang, Zhiguang & Yan, Weizhong & Oates, Tim. (2017). Time series classification from scratch with deep neural networks: A strong baseline. 1578-1585. 10.1109/IJCNN.2017.7966039.

6) Ismail Fawaz, Hassan & Forestier, Germain & Weber, Jonathan & Idoumghar, Lhassane & Muller, Pierre-Alain. (2019). Deep learning for time series classification: a review. Data Mining and Knowledge Discovery. 33. 10.1007/s10618-019-00619-1.

Acknowledgements

This work was supported by NIH grants AI116794, DK112217,ES013508, EY022300, HL134015, LM009012, LM010098, LM011360, and TR001263.

Lab/Research Links

www.med.upenn.edu/urbslab/
www.ryanurbanowicz.com

Code Availability

Github.com/UrbsLab