



# Music Classification According To Composers

Michael Raafat, Amr Nasr

## Faculty Of Engineering, Computer and Systems Department

### Objectives

Build classification model for music that can be generalized for other music recognition tasks.

Trying many approaches for Music Classification

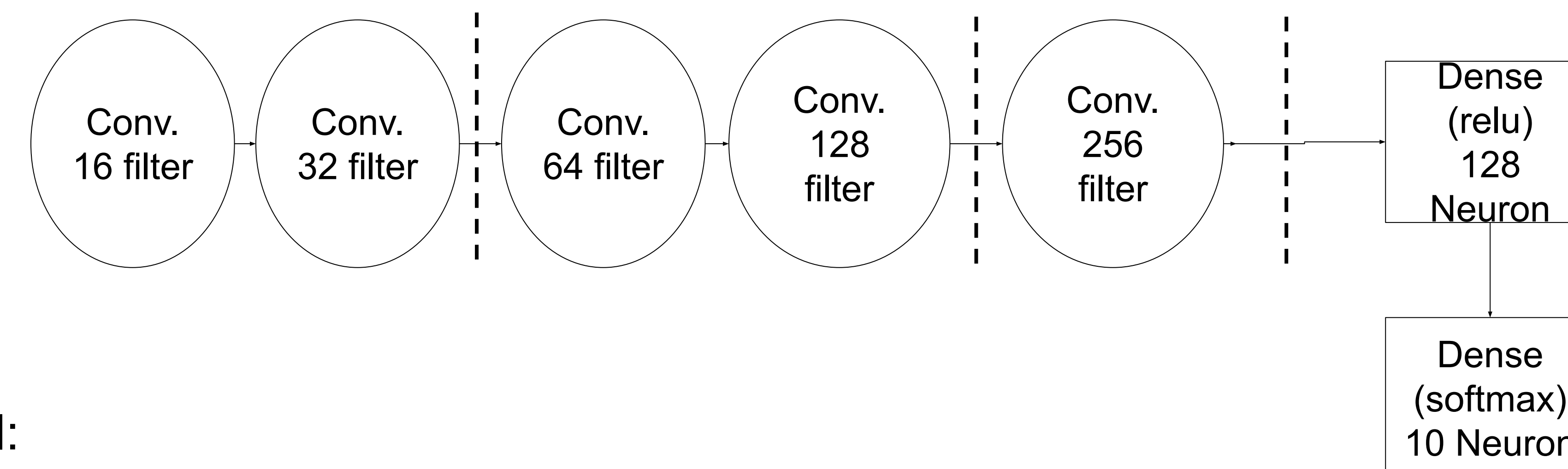
### Data Preprocessing

- We used the pretty midi python library to parse and act as a wrapper to the midi files.
- We extracted from each midi file its piano roll with a sample frequency of 10, the piano roll is a matrix representing the volume of each musical note across the different timesteps flattened across instruments.
- We filtered any tracks that are less than 30 second.
- We separated our dataset to training, validation and testing with 70%, 15% and 15% respectively.
- We augmented the separated dataset using a window approach, taking multiple random windows of 30 second from each track (normally 60 from each track), which helped as balance the dataset since the tracks number were almost balanced already

### Models

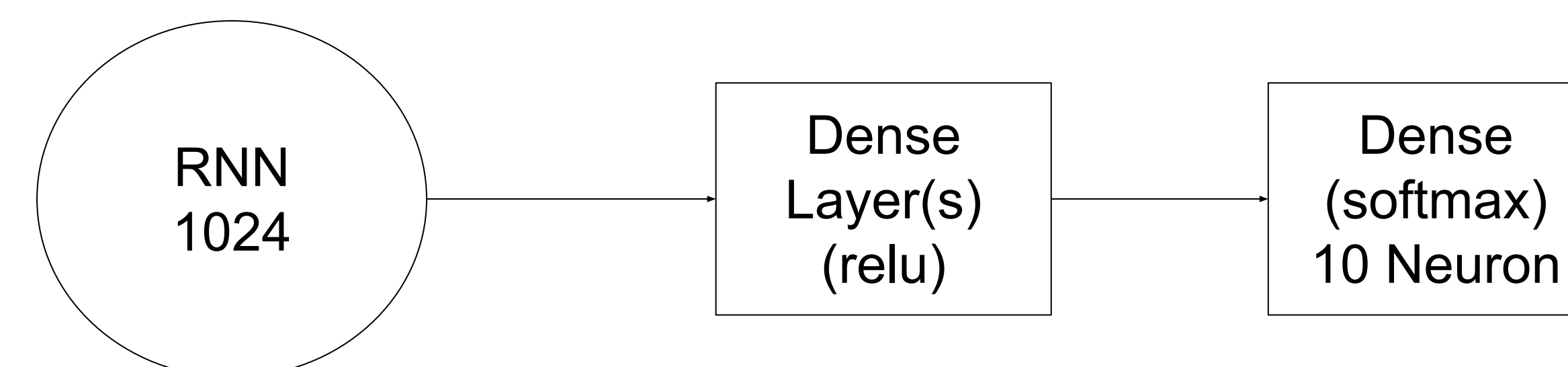
CNN Model:

- Default convolutional layers kernel sizes are 3x3.
- After each 2 convolutional layer, there is a max pooling layer.
- After each convolutional layer, there is a relu activation and batch normalization



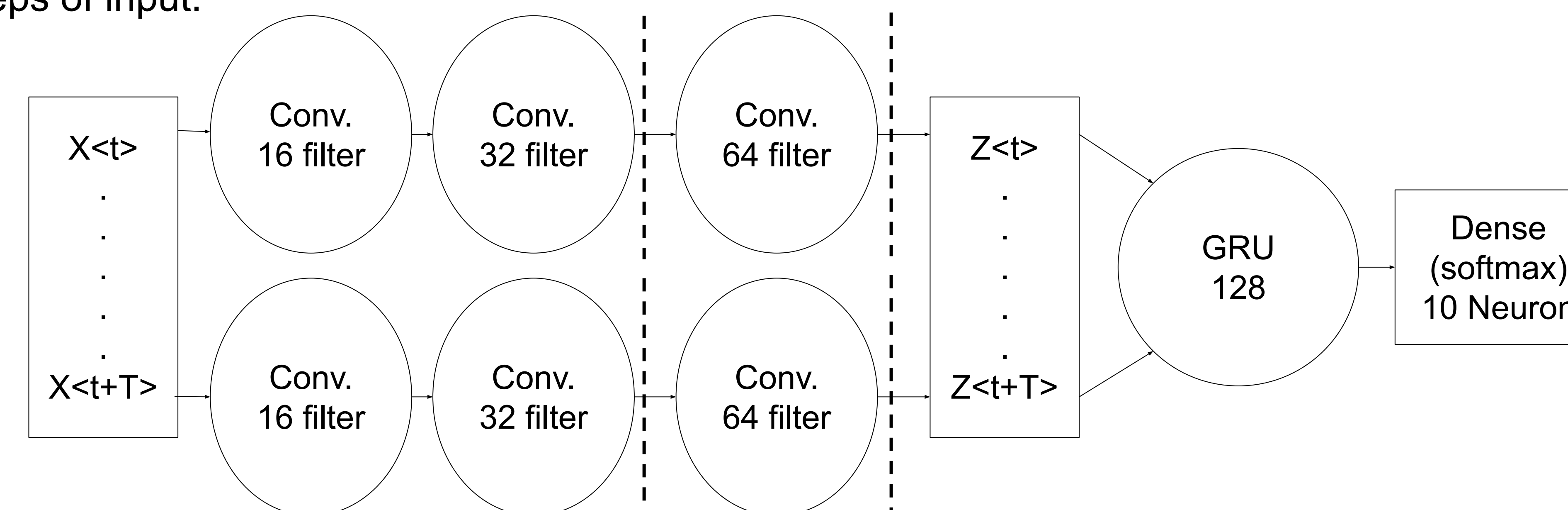
RNN Model:

- One LSTM or GRU with 1024 units.
- Dense layer(s).



Hybrid Model:

- One LSTM or GRU with 128 units.
- All convolutional layers kernel sizes are 5x5.
- Each dotted line represents a max pooling layer
- After each convolutional layer, there is a relu activation and batch normalization.
- All convolutions are 1D, the convolutional model has parameter tying and is the same that is applied on all timesteps of input.



### Results

- CNN Model:  
82.51% Single Accuracy  
92.68% Voting Accuracy
- RNN Model:  
59.58% Single Accuracy  
74.39% Voting Accuracy
- Hybrid Model:  
73.15% Single Accuracy  
81.70% Voting Accuracy

### Conclusion

- CNN Architecture is the best of all models.
- Colab Limitations limits our ability to test more complex models especially in hybrid builds.
- A composer fingerprint is present and can be obtained from 30 seconds of music, this opens up a lot of possible usage.

### Future Work

- Since results can be obtained on a fixed set of composers, we want to generalize this concept by trying to apply siamese networks as one shot learning using our best model as the feature extractor.
- Change voting method to a weighted voting mechanism using the probabilities produced by the network.

### References

- <https://github.com/chramsey/CS230-Final-Project>
- <http://www.midiworld.com/classic.htm>
- <http://www.piano-midi.de/>