A study on effective music genre classification based on multiple spectral features using con volutional neural networks

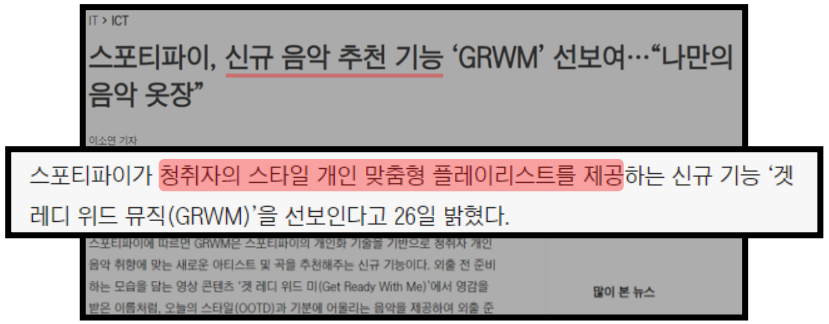
Sung-Hyun Cho

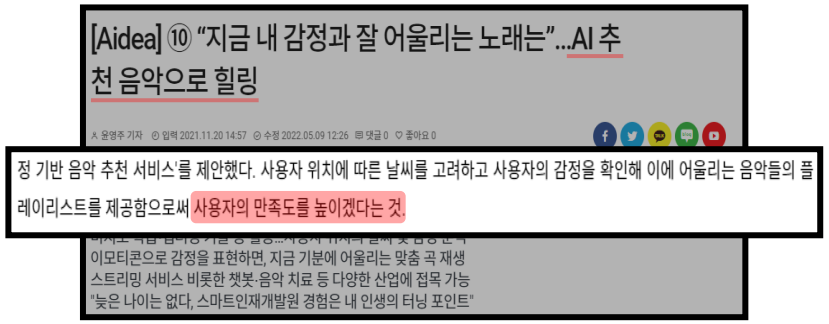
saintcho94@gmail.com

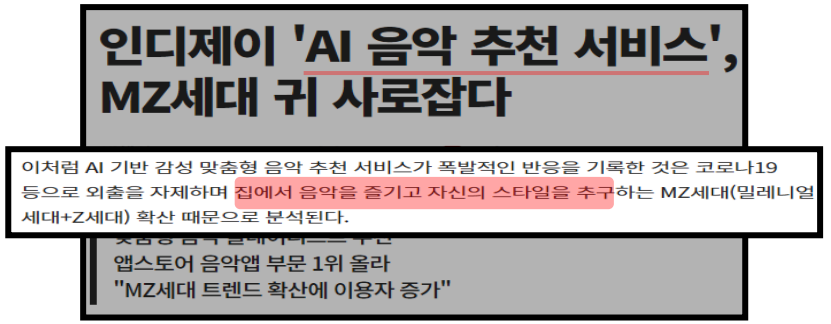
08th Nov, 2022

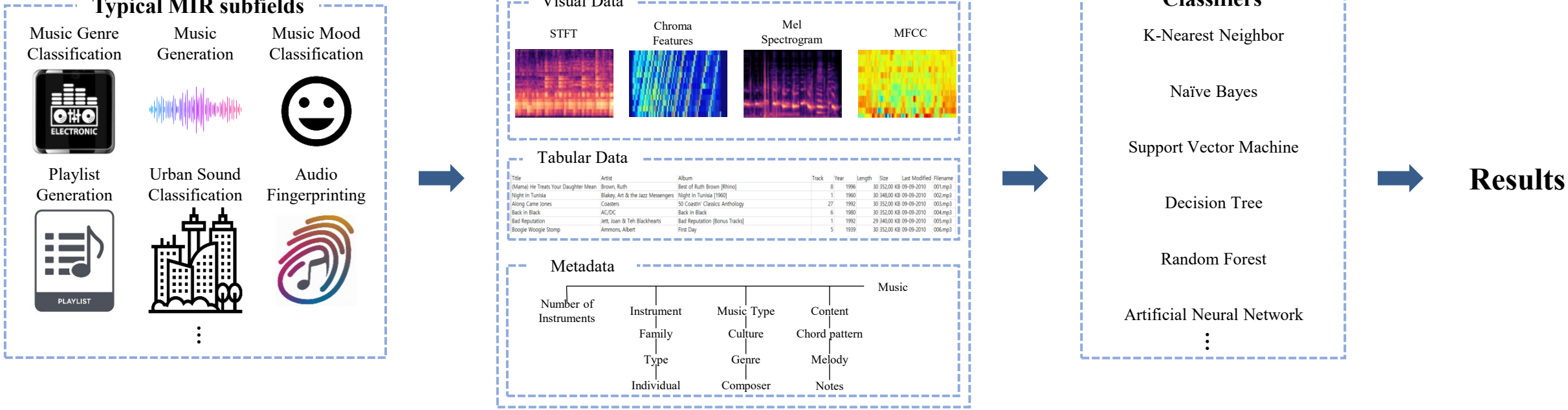
Music Information Retrieval (MIR)

- Concerned with the **extraction** and **inference of meaningful features** from music.
- Aims at **making the vast store of music** available to individuals.
- Different representations of **music-related subjects** and **items are considered**.



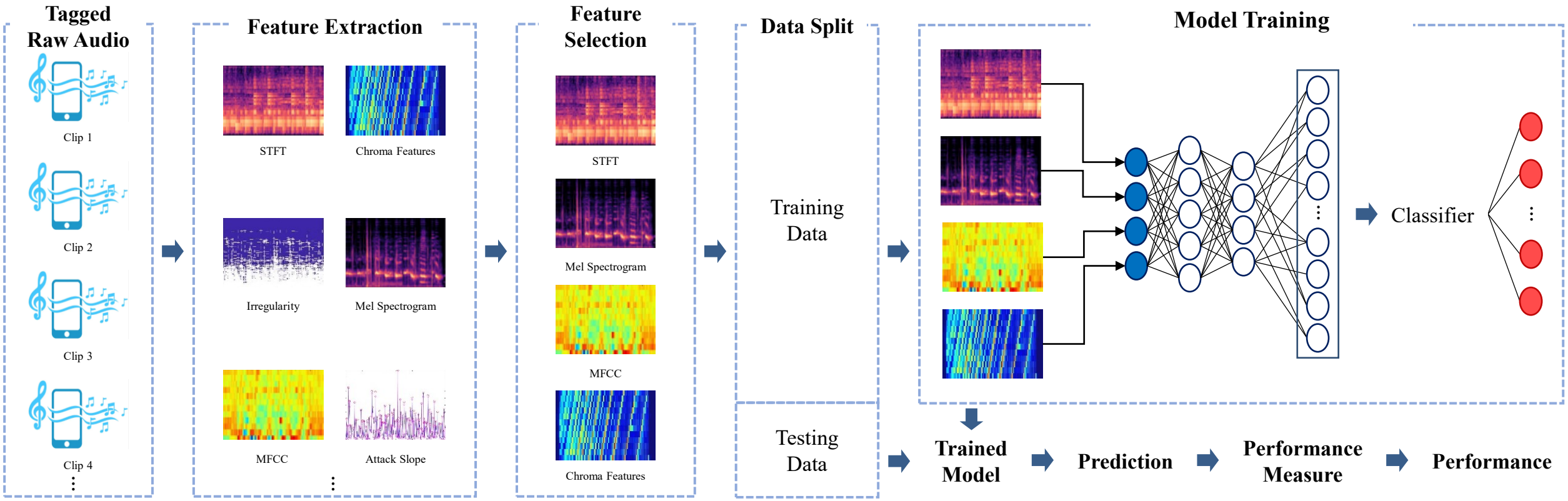






Music Genre Classification Based on Convolutional Neural Network

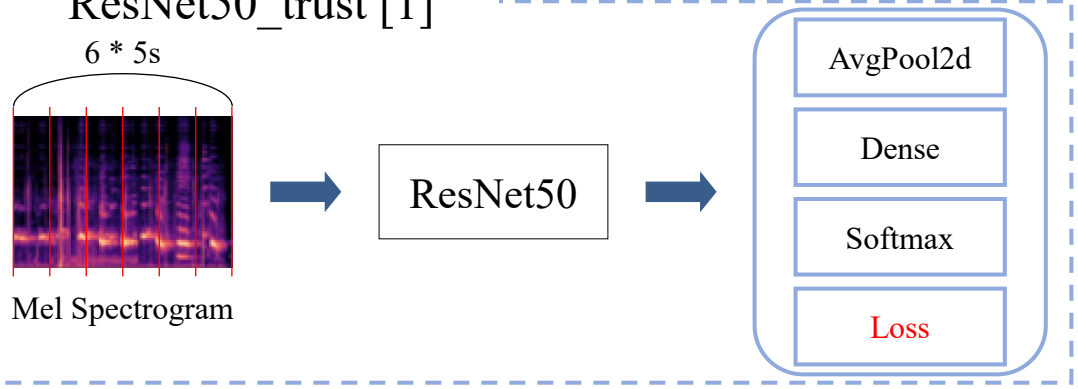
- **Extract features** from tagged raw audio.
- **Select features** from extracted features and separate into training and testing data.
- **Training data** as **model training**
- **Testing data** as **performance measure**.



Related Work

Comparison Model Structures

ResNet50_trust [1]



$$\text{Loss} = (1-t)\cdot Loss1 + t\cdot Loss2$$

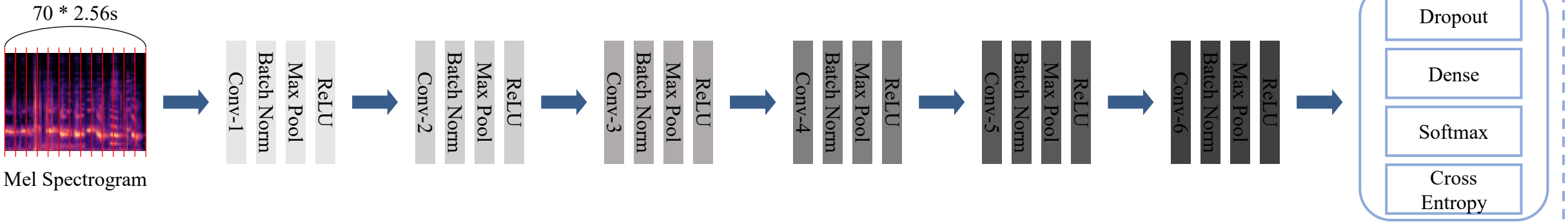
$t = a^{-b}$ $(a: balance\ base, b: balance\ factor)$

$\text{Loss1} = \text{Cross\_Entropy}(y_{\text{true}}(\text{Onehot encoding}), y_{\text{pred}}(\text{Softmax function}))$

$\text{Loss2} = \text{Cross\_Entropy}(y_{\text{true}}(\text{Uniform Distribution function}), y_{\text{pred}}(\text{Softmax function}))$

- The performance was improved with **few modifications** to the existing model.
- The data was **arbitrarily modified** during the data preprocessing process.

Pelchat & Gelowitz [2]

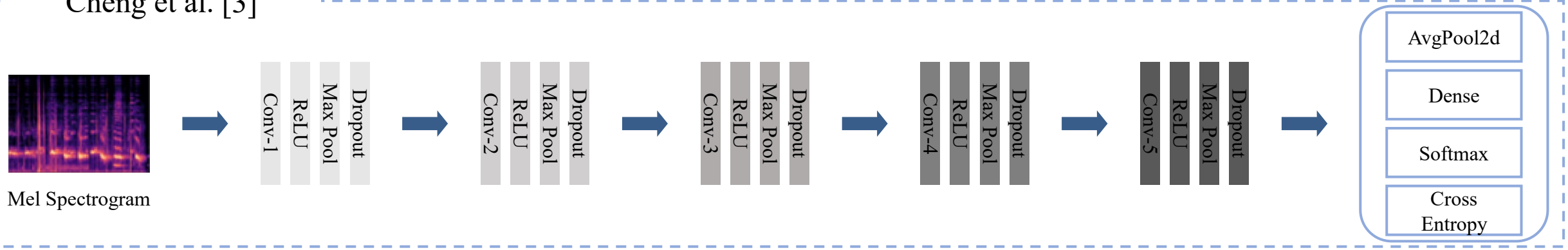


- The model showed comparable performance with **a short music clip**.
- Performance declined by **training even the meaningless parts** of the classification.

Related Work

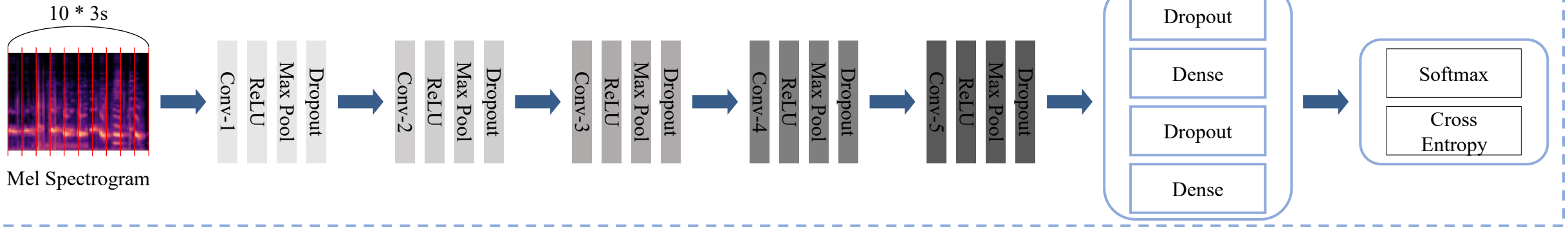
Comparison Model Structures

Cheng et al. [3]



- Several classifiers performed classification through **majority voting** to increase accuracy.
- It requires a long experimental time of **13.1 hours**.

Shah et al. [4]

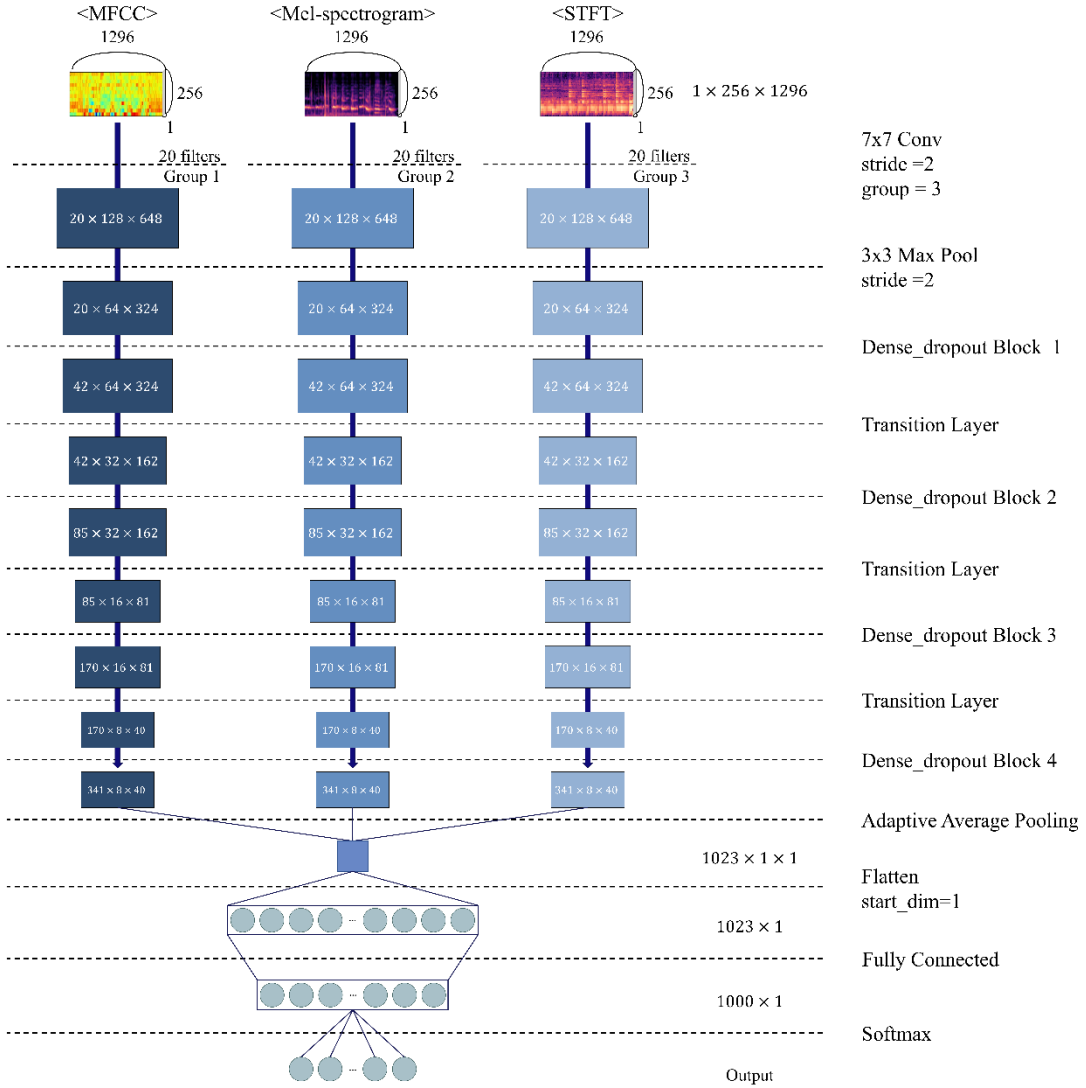


- Achieved high classification accuracy by **designing a new convolutional block**.
- By comparing the model with conventional machine learning methods, it is not known **whether it is superior** to other CNN models.

Proposed Method

Model Flowchart

Model Flowchart

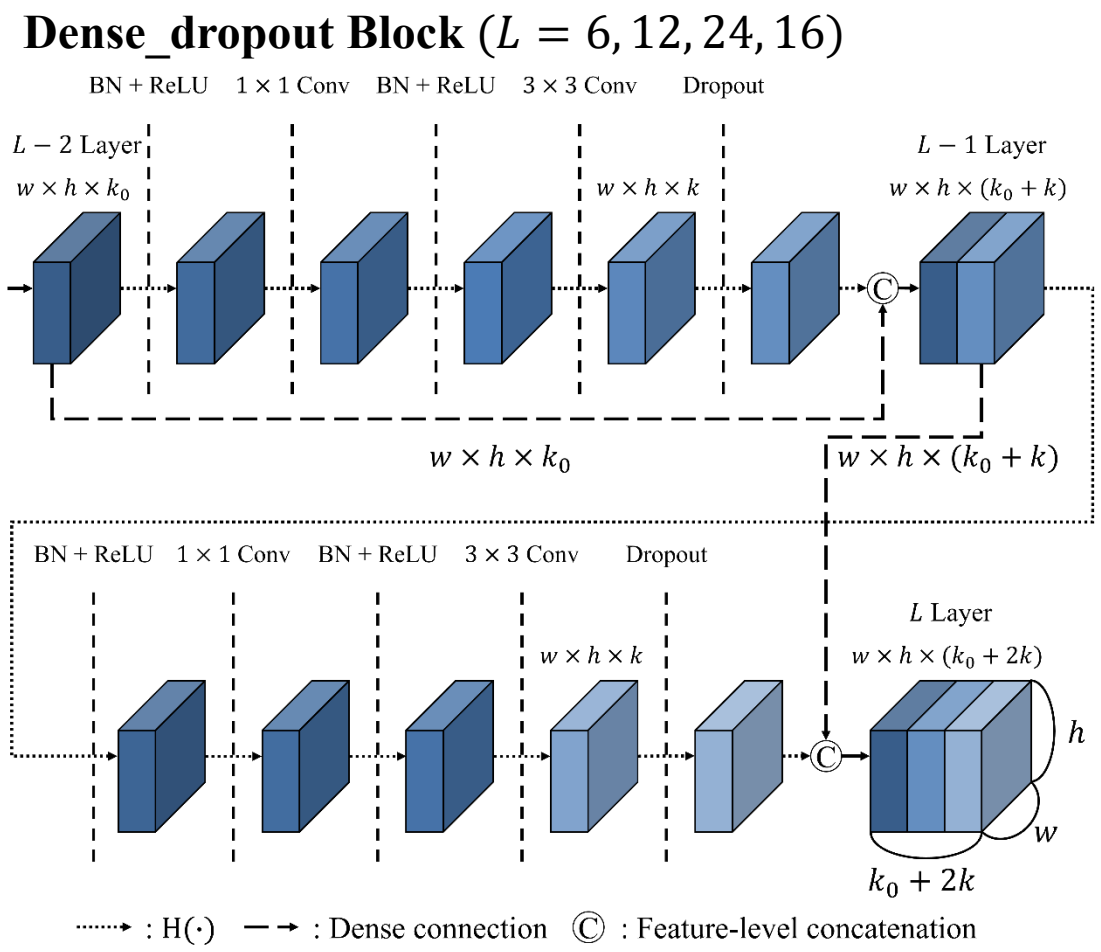


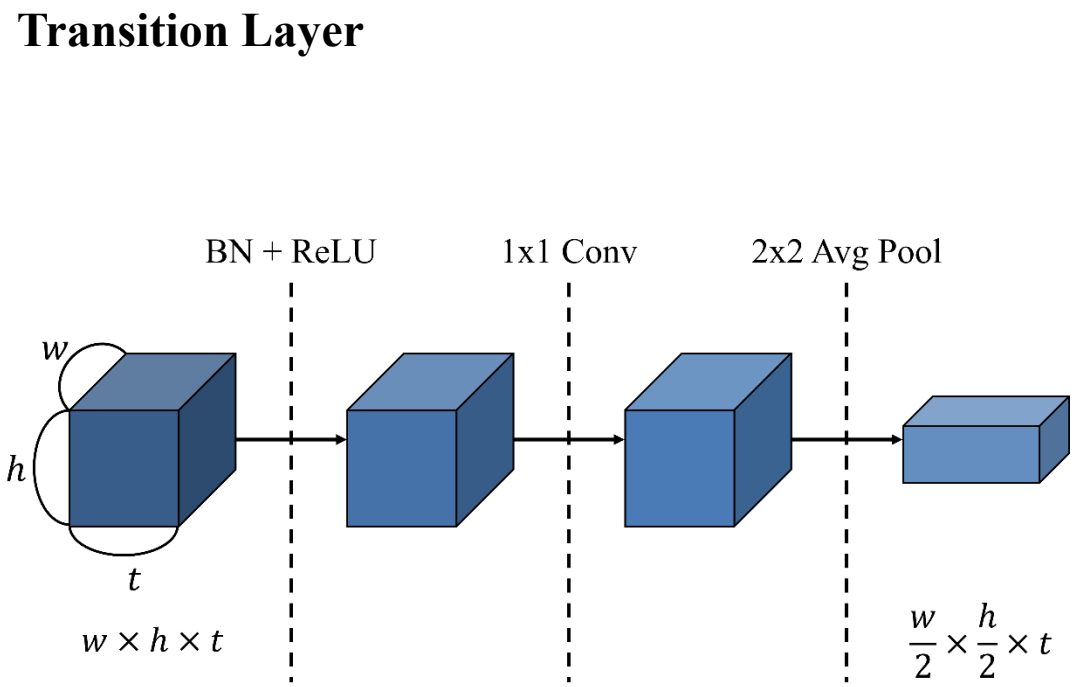
Model Architecture

Layers	Output Feature Map Size per Channel			Convolution and Pooling Layers
Initial Convolution	(20, 128, 648)	(20, 128, 648)	(20, 128, 648)	7×7 *conv* , stride = 2, group = 3
	(20, 64, 324)	(20, 64, 324)	(20, 64, 324)	Max Pooling, stride = 2
Dense_dropout Block 1	(42, 64, 324)	(42, 64, 324)	(42, 64, 324)	$\begin{bmatrix} 1 \times 1 \ conv \\ 3 \times 3 \ conv \end{bmatrix} \times 6$
Transition Layer	(42, 64, 324)	(42, 64, 324)	(42, 64, 324)	Convolution
	(42, 32, 162)	(42, 32, 162)	(42, 32, 162)	Average Pooling, stride = 2
Dense_dropout Block 2	(85, 32, 162)	(85, 32, 162)	(85, 32, 162)	$\begin{bmatrix} 1 \times 1 \ conv \\ 3 \times 3 \ conv \end{bmatrix} \times 12$
Transition Layer	(85, 32, 162)	(85, 32, 162)	(85, 32, 162)	Convolution
	(85, 16, 81)	(85, 16, 81)	(85, 16, 81)	Average Pooling, stride = 2
Dense_dropout Block 3	(170, 16, 81)	(170, 16, 81)	(170, 16, 81)	$\begin{bmatrix} 1 \times 1 \ conv \\ 3 \times 3 \ conv \end{bmatrix} \times 24$
Transition Layer	(170, 16, 81)	(170, 16, 81)	(170, 16, 81)	Convolution
	(170, 8, 40)	(170, 8, 40)	(170, 8, 40)	Average Pooling, stride = 2
Dense_dropout Block 4	(341, 8, 40)	(341, 8, 40)	(341, 8, 40)	$\begin{bmatrix} 1 \times 1 \ conv \\ 3 \times 3 \ conv \end{bmatrix} \times 16$
Classification Layer	(1023, 1, 1)			Global Average Pooling
				1000D Fully-Connected, Softmax

Proposed Method

Model Flowchart





Experimental Settings

Datasets

- Data Description
  - ➢ **12 music genre datasets** from the creation stage and by post-processing

Dataset	Patterns	Avg. Length	# of Classes	Average Patterns per Class	±	Standard Deviation
Ballroom	698	30	8	87.250	±	17.555
Ballroom-Extended	4180	30	13	321.54	±	195.70
emoMusic-G	744	45	8	93.00	±	16.1776
Emotify-G	400	43	4	100.00	±	0.00
FMA-SMALL	7996	30	8	999.500	±	0.500
GiantStepsKey-G	604	80	23	24.9130	±	28.0274
GMD-G	1042	300	18	57.89	±	70.07
GTZAN	1000	30	10	100.00	±	0.00
HOMBURG	1886	10	9	209.56	±	134.87
ISMIR04	727	120	6	121.167	±	95.602
MICM	1137	360	7	162.429	±	119.717
Seyerlehner-Unique	3115	30	10	311.500	±	248.349

Experimental Settings

Hyperparameter Settings of the Proposed Model

Model	Batch Size	Epochs	Loss Function	Optimizer	Learning Rate	Weight Decay
Proposed	16	60	Cross Entropy Loss	Adam	1e-3	1e-4

Hyperparameter Settings of the Comparison Models

Model	Batch Size	Epochs	Loss Function	Optimizer	Learning Rate	Weight Decay
ResNet50_trust [1]	64	60	Proposed Loss Function	Adam	1e-3	1e-4
Pelchat and Gelowitz [2]	128		Cross Entropy Loss			
Cheng et al. [3]	32		Cross Entropy Loss			
Shah et al. [4]	32		Cross Entropy Loss			

Experimental Settings

Cross-Validation for Test Performance

- Number of Iterations
  - ➢ **10** iterations for each dataset
- Data Split Ratio
  - ➢ Train : Test = **0.8 : 0.2**
  - ➢ **Randomly** divided for each iteration

Performance Measure

$$Accuracy(\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100$$

$TP : True\ Positive \quad TN : True\ Negative$

$FP : False\ Positive \quad FN : False\ Negative$

Evaluates the overall effectiveness of a model

➡ **Higher value, Higher Performance**

Experimental Results

Comparison Between the Proposed Late Fusion Method and the Early Fusion Method

▼ : 95% significance level

Dataset	Proposed	Early Fusion Method
Ballroom	**58.29 ± 1.91** ∨	54.57 ± 5.22 ▼
Ballroom-Extended	**85.49 ± 1.06** ∨	82.94 ± 1.42 ▼
emoMusic-G	**36.04 ± 0.95** ∨	35.97 ± 2.29
Emotify-G	**74.88 ± 2.53** ∨	72.88 ± 3.87
FMA-SMALL	**57.33 ± 0.94** ∨	55.63 ± 1.05 ▼
GiantStepsKey-G	**37.27 ± 1.80** ∨	32.81 ± 2.76 ▼
GMD-G	**67.13 ± 1.13** ∨	62.11 ± 2.69 ▼
GTZAN	**82.00 ± 2.24** ∨	81.70 ± 1.53
HOMBURG	**58.76 ± 1.15** ∨	55.37 ± 2.85 ▼
ISMIR04	**80.59 ± 1.41** ∨	79.38 ± 1.07 ▼
MICM	**43.25 ± 2.32** ∨	40.61 ± 2.06 ▼
Seyerlehner-Unique	**72.57 ± 1.19** ∨	70.48 ± 0.77 ▼
Total Win/Tie/Lose	**9/3/0**	0/3/9
Avg . Rank	**1.00** ∨	2.00

Experimental Results

Model Comparison Based on Classification Accuracy

▼** : 99% significance level
▼* : 95% significance level

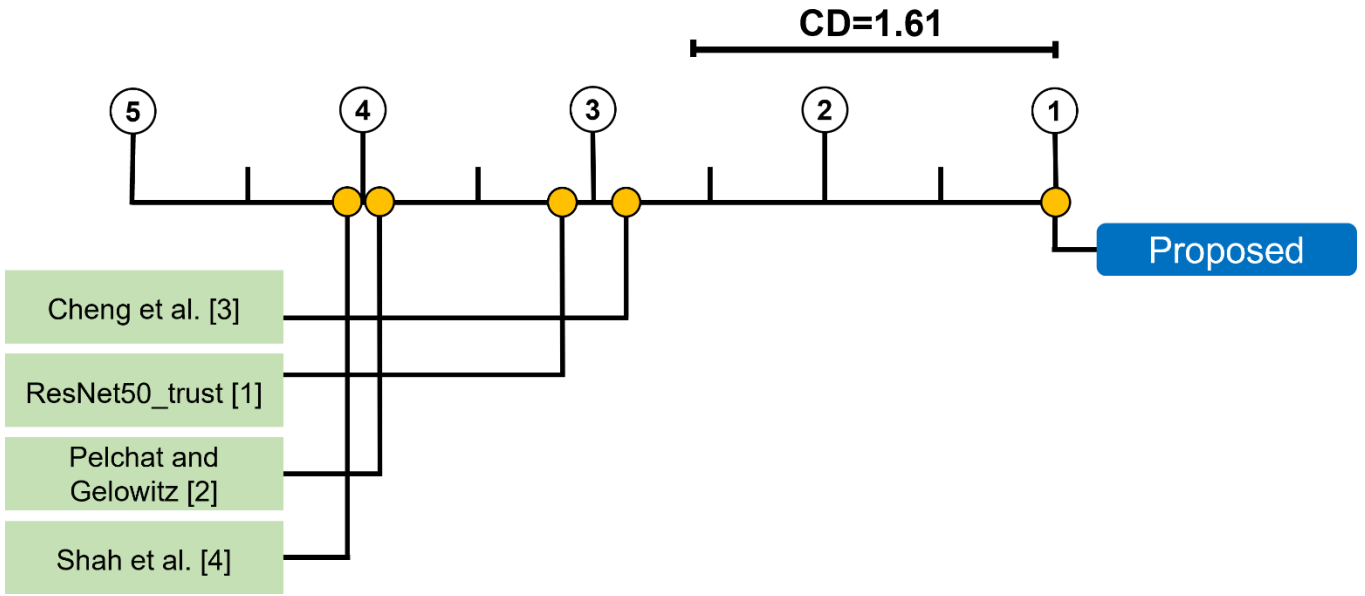
Dataset	Proposed	ResNet50_trust [1]	Pelchat and Gelowitz [2]	Cheng et al. [3]	Shah et al. [4]
Ballroom	**58.29 ± 1.91** ∨	45.71 ± 2.86 ▼**	42.43 ± 2.29 ▼**	48.07 ± 4.79 ▼**	47.71 ± 4.58 ▼**
Ballroom-Extended	**85.49 ± 1.06** ∨	61.82 ± 3.79 ▼**	44.68 ± 1.30 ▼**	62.88 ± 4.70 ▼**	54.38 ± 5.33 ▼**
emoMusic-G	**36.04 ± 0.95** ∨	27.72 ± 3.50 ▼**	28.99 ± 1.78 ▼**	32.95 ± 2.92 ▼**	31.34 ± 3.66 ▼**
Emotify-G	**74.88 ± 2.53** ∨	62.78 ± 2.70 ▼**	67.50 ± 4.17 ▼**	66.00 ± 3.32 ▼**	65.88 ± 5.62 ▼**
FMA-SMALL	**57.33 ± 0.94** ∨	47.03 ± 1.31 ▼**	42.19 ± 1.16 ▼**	46.96 ± 1.62 ▼**	40.88 ± 1.73 ▼**
GiantStepsKey-G	**37.27 ± 1.80** ∨	28.43 ± 3.15 ▼**	25.21 ± 2.53 ▼**	27.19 ± 2.90 ▼**	24.30 ± 2.11 ▼**
GMD-G	**67.13 ± 1.13** ∨	61.91 ± 3.04 ▼**	63.44 ± 1.71 ▼**	39.38 ± 3.36 ▼**	35.36 ± 6.35 ▼**
GTZAN	**82.00 ± 2.24** ∨	77.50 ± 3.81 ▼*	63.35 ± 2.08 ▼**	75.10 ± 3.84 ▼**	72.50 ± 2.13 ▼**
HOMBURG	**58.76 ± 1.15** ∨	53.49 ± 1.59 ▼**	51.01 ± 1.51 ▼**	52.54 ± 2.55 ▼**	48.15 ± 2.14 ▼**
ISMIR04	**80.59 ± 1.41** ∨	70.96 ± 2.93 ▼**	67.88 ± 1.70 ▼**	71.32 ± 1.37 ▼**	67.67 ± 1.99 ▼**
MICM	**43.25 ± 2.32** ∨	37.81 ± 3.32 ▼**	39.30 ± 2.21 ▼**	38.42 ± 3.16 ▼**	39.34 ± 2.28 ▼**
Seyerlehner-Unique	**72.57 ± 1.19** ∨	63.62 ± 2.56 ▼**	60.72 ± 2.73 ▼**	63.50 ± 2.79 ▼**	61.62 ± 1.70 ▼**
Total Win/Tie/Lose	**48/0/0**	16/16/16	3/15/30	15/19/14	5/16/27
Avg . Rank	**1.00** ∨	3.17	3.92	2.83	4.08

Experimental Results

Friedman Test

Evaluation Measure	F_F	Critical Value($\alpha = 0.05$)
Accuracy	29.133	2.498

Bonferroni-Dunn Test



<Accuracy>

Conclusion

Conclusion

- A CNN to effectively perform **music genre classification** in **multiple spectral input situations** was designed.
- The experimental result show that **the proposed method** achieved **higher classification accuracy** than conventional methods using a single input.
- The **late fusion method** may perform **better** than the early fusion method depending on the type of data.

Contribution

- In International Publication (SCI(E))
  - Toward a Fair Evaluation and Analysis of Feature Selection for Music Tag Classification, ***IEEE Access,*** 2021, **Co-author**
- In International Conferences
  - Effective Music Genre Classification using Late Fusion Convolutional Neural Network with Multiple Spectral Features ***ICCE-Asia***, 2022, **1^{st} author**
  - An Efficient Neural Network based on Early Compression of Sparse CT Slice Images, ***PlatCon***, 2021, **Co-author**
- In Domestic Conferences
  - Music Genre Classification Using Multiple Musical Visual Features, ***IEIE***, 2022 , **1^{st} author**
  - Music Genre Classification using CNN architecture with feature concatenation, ***KISS Autumn Conf***, 2021, **Co-author**

Thank you

References

[1] Li, J., Han, L., Li, X., Zhu, J., Yuan, B., & Gou, Z. (2022). An evaluation of deep neural network models for music classification using spectrograms. Multimedia Tools and Applications, 81(4), 4621-4647.

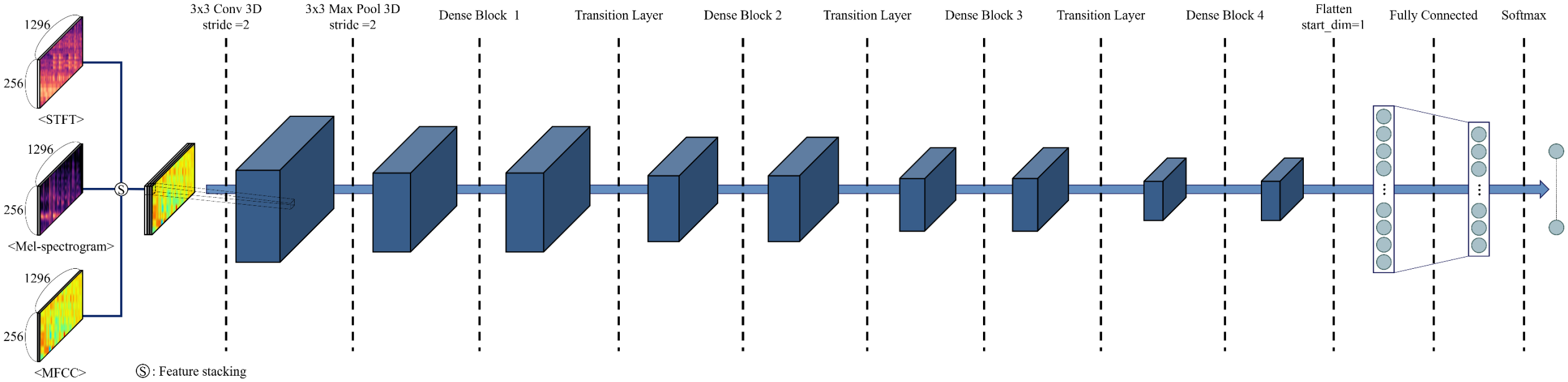
[2] Pelchat, N., & Gelowitz, C. M. (2020). Neural network music genre classification. Canadian Journal of Electrical and Computer Engineering, 43(3), 170-173.

[3] Cheng, Y. H., Chang, P. C., & Kuo, C. N. (2020, November). Convolutional neural networks approach for music genre classification. In 2020 International Symposium on Computer, Consumer and Control (IS3C) (pp. 399-403). IEEE.

[4] Shah, M., Pujara, N., Mangaroliya, K., Gohil, L., Vyas, T., & Degadwala, S. (2022, March). Music Genre Classification using Deep Learning. In 2022 6th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 974-978). IEEE.

Appendix

Appendix: Early Fusion Model Flowchart



Appendix: Calculation of the Size of the Feature Maps

The number of input feature maps of l^{th} layer
- $k_0 + k \times (l-1) \quad (k_0 = 60, k = 32)$

Layer 1 $= 60 + 32 \times (1-1) = 60$
Layer 2 $= 60 + 32 \times (2-1) = 60 + 32 = 92$
Layer 3 $= 60 + 32 \times (3-1) = 60 + 64 = 124$
$\vdots$

Conv2d

$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times padding[0] - dilation[0] \times (kernel\_size[0] - 1) - 1}{stride[0]} + 1 \right\rfloor$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 \times padding[1] - dilation[1] \times (kernel\_size[1] - 1) - 1}{stride[1]} + 1 \right\rfloor$$

AvgPool2d

$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times padding[0] - kernel\_size[0]}{stride[0]} + 1 \right\rfloor$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 \times padding[1] - kernel\_size[1]}{stride[1]} + 1 \right\rfloor$$

MaxPool2d

$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times padding[0] - dilation[0] \times (kernel\_size[0] - 1) - 1}{stride[0]} + 1 \right\rfloor$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 \times padding[1] - dilation[1] \times (kernel\_size[1] - 1) - 1}{stride[1]} + 1 \right\rfloor$$

Appendix: Model Summary

Number of Trainable Parameters = (Input Number of Channels) × (Kernel Size) × (Output Number of Channels)
1x1 conv 60x1x1x128 = 7680
3x3 conv 128x3x3x32 = 36864

	First Convolution		Dense Block 1												Transition Layer	
			Dense Layer 1		Dense Layer 2		Dense Layer 3		Dense Layer 4		Dense Layer 5		Dense Layer 6			
Layers	7x7 Conv	3x3 MaxPool	1x1 Conv	3x3 Conv	1x1 Conv	3x3 Conv	1x1 Conv	3x3 Conv	1x1 Conv	3x3 Conv	1x1 Conv	3x3 Conv	1x1 Conv	1x1 Conv	1x1 Conv	2x2 AvgPool
Input Number of Channels	3	-	60	128	92	128	124	128	156	128	188	128	220	128	252	-
Output Number of Channels	60	-	128	32	128	32	128	32	128	32	128	32	128	32	126	-
Number of Trainable Parameters	2,940	-	7,680	36,864	11,776	36,864	15,872	36,864	19,968	36,864	24,064	36,864	28,160	36,864	31,752	-
Output Image Shape	[3, 256, 1296]	[60, 128, 648]	[60, 64, 324]	[128, 64, 324]	[92, 64, 324]	[128, 64, 324]	[124, 64, 324]	[128, 64, 324]	[156, 64, 324]	[128, 64, 324]	[188, 64, 324]	[128, 64, 324]	[220, 64, 324]	[128, 64, 324]	[252, 64, 324]	[126, 64, 324]

	Dense Block 2												Transition Layer	
	Dense Layer 1	Dense Layer 2	Dense Layer 3	Dense Layer 4	Dense Layer 5	Dense Layer 6	Dense Layer 7	Dense Layer 8	Dense Layer 9	Dense Layer 10	Dense Layer 11	Dense Layer 12		
Layers	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	2x2 AvgPool
Input Number of Channels	126	158	190	222	254	286	318	350	382	414	446	478	510	-
Output Number of Channels	128	128	128	128	128	128	128	128	128	128	128	128	255	-
Number of Trainable Parameters	16,128	20,224	24,320	28,416	32,512	36,608	40,704	44,800	48,896	52,992	57,088	61,184	130,050	-
Output Image Shape	[126, 32, 162]	[158, 32, 162]	[190, 32, 162]	[222, 32, 162]	[254, 32, 162]	[286, 32, 162]	[318, 32, 162]	[350, 32, 162]	[382, 32, 162]	[414, 32, 162]	[446, 32, 162]	[478, 32, 162]	[510, 32, 162]	[255, 32, 162]

Appendix: Model Summary

	Dense Block 3																							
	Dense Layer 1	Dense Layer 2	Dense Layer 3	Dense Layer 4	Dense Layer 5	Dense Layer 6	Dense Layer 7	Dense Layer 8	Dense Layer 9	Dense Layer 10	Dense Layer 11	Dense Layer 12	Dense Layer 13	Dense Layer 14	Dense Layer 15	Dense Layer 16	Dense Layer 17	Dense Layer 18	Dense Layer 19	Dense Layer 20	Dense Layer 21	Dense Layer 22	Dense Layer 23	Dense Layer 24	Transition Layer	
Layers	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x2 Conv	1x2 Conv	1x1 Conv	1x3 Conv	1x3 Conv	1x1 Conv	1x4 Conv	1x4 Conv	1x1 Conv	1x2 Conv	1x2 Conv	1x1 Conv	1x1 Conv	2x2 AvgPool
Input Number of Channels	255	287	319	351	383	415	447	479	511	543	575	607	639	671	703	735	767	799	831	863	895	927	959	991	1023	
Output Number of Channels	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	511	-
Number of Trainable Parameters	32640	36736	40832	44928	49024	53120	57216	61312	65408	69504	73600	77696	81792	85888	89984	94080	98176	102272	106368	110464	114560	118656	122752	126848	522753	-
Output Image Shape	[255, 16, 81]	[287, 16, 81]	[319, 16, 81]	[351, 16, 81]	[383, 16, 81]	[415, 16, 81]	[447, 16, 81]	[479, 16, 81]	[511, 16, 81]	[543, 16, 81]	[575, 16, 81]	[607, 16, 81]	[639, 16, 81]	[671, 16, 81]	[703, 16, 81]	[735, 16, 81]	[767, 16, 81]	[799, 16, 81]	[831, 16, 81]	[863, 16, 81]	[895, 16, 81]	[927, 16, 81]	[959, 16, 81]	[991, 16, 81]	[1023, 16, 81]	[511, 16, 81]

	Dense Block 4														
	Dense Layer 1	Dense Layer 2	Dense Layer 3	Dense Layer 4	Dense Layer 5	Dense Layer 6	Dense Layer 7	Dense Layer 8	Dense Layer 9	Dense Layer 10	Dense Layer 11	Dense Layer 12	Dense Layer 13	Dense Layer 14	Dense Layer 15	Dense Layer 16
Layers	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x1 Conv	1x2 Conv	1x2 Conv	1x1 Conv	1x3 Conv
Input Number of Channels	511	543	575	607	639	671	703	735	767	799	831	863	895	927	959	991
Output Number of Channels	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128
Number of Trainable Parameters	65408	69504	73600	77696	81792	85888	89984	94080	98176	102272	106368	110464	114560	118656	122752	126848
Output Image Shape	[511, 8, 40]	[543, 8, 40]	[575, 8, 40]	[607, 8, 40]	[639, 8, 40]	[671, 8, 40]	[703, 8, 40]	[735, 8, 40]	[767, 8, 40]	[799, 8, 40]	[831, 8, 40]	[863, 8, 40]	[895, 8, 40]	[927, 8, 40]	[959, 8, 40]	[991, 8, 40]