EE798R: PHASE 2 RESULTS

Karan Mundhra
Indian Institute of Technology Kanpur
karanm21@iitk.ac.in

November 6, 2024

1 Architecture Modifications for LIGHT-SERNET

This report outlines modifications to LIGHT-SERNET for enhanced Speech Emotion Recognition (SER) performance. I have used mainly five modifactions. The results are also shown corresponding to each modification.

Background and Original Model Overview

The original LIGHT-SERNET is a lightweight fully convolutional neural network designed for Speech Emotion Recognition (SER) on devices with limited resources. It is structured with:

1. **Input Pipeline:** Extracts MFCC features, allowing effective preprocessing for SER.
2. **Body Part I (Parallel Convolutions):** Uses three parallel convolution paths with different kernel sizes to balance spectral and temporal information.
3. **Body Part II (Feature Learning Blocks):** Sequential layers capture higher-level representations.
4. **Head:** Includes a dropout and fully-connected softmax layer for classification.

1.1 Modification 1: Selective Attention in Paths

1.1.1 Channel Attention in Temporal Path (1×11 Convolution)

Purpose: Emphasizes channels capturing time-dependent emotional features.
Implementation: Channel Attention was added after the 1×11 convolution in the temporal path. This layer applies learned weights to channels based on their relevance over time.

Algorithm 1 Channel Attention

$avg\_pool \leftarrow GlobalAveragePooling2D(input)$
$max\_pool \leftarrow GlobalMaxPooling2D(input)$
$channel\_weights \leftarrow Dense(sigmoid(avg\_pool + max\_pool))$
return $Multiply(input, channel\_weights)$

1.1.2 Spatial Attention in Spectral-Temporal Path (3×3 Convolution)

Purpose: Focuses on spatial regions with significant emotional content.
Implementation: Spatial Attention added after 3×3 convolution in the spectral-temporal path. It uses average and max pooling to compute a spatial map applied to enhance critical regions.

Algorithm 2 Spatial Attention

$avg\_pool \leftarrow Lambda(mean(input))$
$max\_pool \leftarrow Lambda(max(input))$
$attention\_map \leftarrow Conv2D(sigmoid(concat(avg\_pool, max\_pool)))$
return $Multiply(input, attention\_map)$

1.1.3 Global Channel Attention in Final LFLB (Body Part II)

Purpose: Prioritizes high-level emotional features before classification.
Implementation: A Global Channel Attention layer was added in the final Local Feature Learning Block to emphasize channels that strongly indicate emotional content.

Algorithm 3 Global Channel Attention

$avg\_pool \leftarrow GlobalAveragePooling2D(input)$
$attention\_weights \leftarrow Dense(sigmoid(avg\_pool))$
return $Multiply(input, attention\_weights)$

1.1.4 Results

The modified model demonstrated an improvement in accuracy metrics. The results for the IEMOCAP dataset (scripted+improvised, 7 seconds, with Focal Loss) are summarized below.

result > IEMOCAP_7.0s_Segmented_focal_Report_attention_all.txt

	precision	recall	f1-score	support
angry	0.7600	0.7250	0.7420	1103
sadness	0.7250	0.7550	0.7397	1084
neutral	0.6600	0.7100	0.6842	1708
happiness_excited	0.6850	0.6675	0.6762	1636
accuracy			0.7140	5531
macro avg	0.7075	0.7144	0.7105	5531
weighted avg	0.7155	0.7140	0.7132	5531

Figure 1: Performance metrics for modified LIGHT-SERNET on IEMOCAP dataset

Table 1: Summary of Metrics for IEMOCAP (scripted+improvised, 7 seconds, Focal Loss)

Metric	Value
Unweighted Average Recall (UAR)	71.44%
Weighted Accuracy (WA)	71.40%
F1-score	71.05%

These seclective attention paths improved the net performance of the model. But, there was a trade-off also. Then model size became 1.07 MB.

1.2 Modification 2: Attention Mechanism in Initial and Mid-Layers

1.2.1 Hybrid Attention in Initial Convolutional Paths

Purpose: Enhance feature diversity by combining local (convolutional) and global (attention) features.
Implementation: Added in Body Part I after the input layer. Alongside the temporal (1×11), spectral (1×9), and spectral-temporal (3×3) convolution paths, a Scaled Dot-Product Attention path was introduced.

1.2.2 Mid-Layer Attention

Purpose: Integrate attention mid-layer to refine features progressively.
Implementation: Added in Body Part II following an intermediate convolution. ‘Query‘, ‘key‘, and ‘value‘ tensors are derived from the feature map, and the attention output is combined with the feature map using a residual connection.

Algorithm 4 Hybrid Attention Block in Initial Paths

$query \leftarrow Conv2D(1,1)(input)$
$key \leftarrow Conv2D(1,1)(input)$
$value \leftarrow Conv2D(1,1)(input)$
$attention\_output, \_ \leftarrow \text{ScaledDotProductAttention}(query, key, value)$
return $Concatenate(path1, path2, path3, attention\_output)$

Algorithm 5 Mid-Layer Attention Block

$query, key, value \leftarrow Conv2D(1,1)(x)$
$attention\_output, \_ \leftarrow \text{ScaledDotProductAttention}(query, key, value)$
return $Add(x, attention\_output)$

1.2.3 Results

The results for the IEMOCAP dataset (scripted+improvised, 7 seconds, with Focal Loss) are summarized below.

result > phase2 > IEMOCAP_7.0s_Segmented_focal_Report_hybrid_all.txt

	precision	recall	f1-score	support
angry	0.7194	0.7344	0.7268	1103
sadness	0.7021	0.7002	0.7012	1084
neutral	0.6204	0.7014	0.6584	1708
happiness_excited	0.6827	0.5813	0.6279	1636
accuracy			0.6722	5531
macro avg	0.6811	0.6793	0.6786	5531
weighted avg	0.6746	0.6722	0.6714	5531

Figure 2: Performance metrics for Hybrid Attention LIGHT-SERNET on IEMOCAP dataset

Table 2: Summary of Metrics for IEMOCAP (scripted+improvised, 7 seconds, Focal Loss)

Metric	Value
Unweighted Average Recall (UAR)	67.93%
Weighted Accuracy (WA)	67.22%
F1-score	67.86%

The hybrid attention modification did not significantly improve performance and increased the model size to 1.15 MB, adding complexity without any good improvement. So, I tried to apply the attention mechanism in a more effective way (Modification 1).

1.3 Modification 3: Depthwise Separable Convolutions

1.3.1 Depthwise Separable Convolutions in Initial Paths

Purpose: Reduce model size and computational cost while retaining performance.
Implementation: Replaced standard convolutions with depthwise separable convolutions in the initial paths (Temporal, Spectral, and Spectral-Temporal). Depthwise separable convolutions split each standard convolution into two steps:

1. **Depthwise Convolution**: Applies a filter to each input channel independently, reducing parameter count.

2. **Pointwise Convolution**: Combines the channels with a 1×1 convolution.

This factorization significantly reduces the number of parameters and floating-point operations.

Algorithm 6 Depthwise Separable Convolution

$path1 \leftarrow SeparableConv2D(32, (11, 1), padding = "same")(input)$
$path2 \leftarrow SeparableConv2D(32, (1, 9), padding = "same")(input)$
$path3 \leftarrow SeparableConv2D(32, (3, 3), padding = "same")(input)$
$feature\_extractor \leftarrow Concatenate(path1, path2, path3)$

1.3.2 Mathematical Justification

Depthwise separable convolutions factorize a convolution into depthwise and pointwise steps. For each convolution with $k \times k$ kernel, they reduce parameters from $k^2 \times D \times D_{out}$ to $k^2 \times D + D \times D_{out}$, resulting in fewer operations.

1.3.3 Results

This modification reduced the model size and computational cost, making it more efficient for lightweight applications. The results for the IEMOCAP dataset (scripted+improvised, 7 seconds, with Focal Loss) are summarized below.

result > phase2 > IEMOCAP_7.0s_Segmented_focal_Report_sepConv_all.txt

	precision	recall	f1-score	support
angry	0.7928	0.7144	0.7515	1103
sadness	0.7192	0.7371	0.7280	1084
neutral	0.6300	0.7237	0.6736	1708
happiness_excited	0.6858	0.6137	0.6477	1636
accuracy			0.6919	5531
macro avg	0.7069	0.6972	0.7002	5531
weighted avg	0.6964	0.6919	0.6922	5531

Figure 3: Performance metrics for Depthwise Separable Convolutions in LIGHT-SERNET on IEMOCAP dataset

Table 3: Summary of Metrics for IEMOCAP (scripted+improvised, 7 seconds, Focal Loss)

Metric	Value
Unweighted Average Recall (UAR)	69.72%
Weighted Accuracy (WA)	69.19%
F1-score	70.02%

This modification reduced the model size from 0.88 MB to 0.80 MB, making it efficient with very slight decrease in the model's performance.

1.4 Modification 4: Skip Connections (Residual Learning)

1.4.1 Skip Connections in Convolutional Layers

Purpose: Facilitate gradient flow in deeper layers, helping prevent vanishing gradients and enhancing training stability.
Implementation: Skip connections were added within the main convolutional paths (Temporal, Spectral, and

Spectral-Temporal) in Body Part I and in the subsequent convolutional layers in Body Part II. Each skip connection adds the input of a convolutional layer back to its output, preserving information and ensuring effective gradient propagation.

Algorithm 7 Skip Connection in Convolutional Layers

$conv\_out \leftarrow Conv2D(input)$
$conv\_out \leftarrow BatchNorm(conv\_out)$
$conv\_out \leftarrow ReLU(conv\_out)$
$skip\_out \leftarrow Conv2D(input)$
$output \leftarrow Add(conv\_out, skip\_out)$
return *output*

1.4.2 Reasoning and Benefits

The addition of skip connections helps retain information across layers, facilitating smoother gradient flow. This mitigates the risk of vanishing gradients, particularly in deeper layers, allowing the model to capture complex features more effectively.

1.4.3 Results

The skip connections improved training stability. The results for the IEMOCAP dataset (scripted+improvised, 7 seconds, with Focal Loss) are summarized below.

result > IEMOCAP_7.0s_Segmented_focal_Report_skip_all.txt

	precision	recall	f1-score	support
angry	0.7831	0.6908	0.7341	1103
sadness	0.6985	0.7887	0.7409	1084
neutral	0.6325	0.7406	0.6823	1708
happiness_excited	0.7264	0.5923	0.6525	1636
accuracy			0.6963	5531
macro avg	0.7101	0.7031	0.7025	5531
weighted avg	0.7033	0.6963	0.6953	5531

Figure 4: Performance metrics for Skip Connections in LIGHT-SERNET on IEMOCAP dataset

Table 4: Summary of Metrics for IEMOCAP (scripted+improvised, 7 seconds, Focal Loss)

Metric	Value
Unweighted Average Recall (UAR)	70.31%
Weighted Accuracy (WA)	69.63%
F1-score	70.25%

The modification increased model size minimally, from 0.88 MB to 0.92 MB, with improved training stability. This is proved as previously in the paper, they trained for 300 epochs and 10 folds. But, with skip conections, I ran the model for 100 epochs and 10 folds, still the model competed with the original model and showed nearly equal performance. It does helped to reduce the training period too.

1.5 Modification 5: LR Warm-Up and Cosine Annealing + Skip Connections

1.5.1 Combined Learning Rate Schedule and Skip Connections

Purpose: Improve training stability and convergence by combining a gradual learning rate schedule with skip connections to enhance gradient flow.
Implementation: This approach combines two techniques:

1. **Learning Rate Warm-Up and Cosine Annealing:** The learning rate starts with a warm-up phase, linearly increasing from a small initial rate (η_{start}) to a target rate (η_{target}) over T_{warmup} epochs:

$$\eta_t = \eta_{start} + (\eta_{target} - \eta_{start}) \times \frac{t}{T_{warmup}}$$

After warm-up, a cosine annealing schedule decreases the learning rate gradually:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{target} - \eta_{min}) \left(1 + \cos\left(\frac{\pi(t - T_{warmup})}{T_{total} - T_{warmup}}\right)\right)$$

2. **Skip Connections:** Added skip connections in main convolutional paths and subsequent layers to facilitate gradient flow, aiding model stability and minimizing vanishing gradient issues.

Algorithm 8 Learning Rate Scheduler with Skip Connections

Input: epoch, initial_lr, min_lr, target_lr, warmup_epochs, total_epochs
if epoch $<$ warmup_epochs **then**
  $lr \leftarrow$ initial_lr + (target_lr − initial_lr) × (epoch/warmup_epochs)
else
  $cosine\_decay \leftarrow 0.5 \times (1 + \cos(\pi \times (\text{epoch} - \text{warmup\_epochs})/(\text{total\_epochs} - \text{warmup\_epochs})))$
  $lr \leftarrow$ min_lr + (target_lr − min_lr) × cosine_decay
end if
return max(lr, min_lr)

1.5.2 Results

The combined learning rate schedule with skip connections contributed to improved training stability and model generalization. The results for the IEMOCAP dataset (scripted+improvised, 7 seconds, with Focal Loss) are summarized below.

result > IEMOCAP_7.0s_Segmented_focal_Report_warmup_all.txt

	precision	recall	f1-score	support
angry	0.7550	0.7350	0.7449	1103
sadness	0.7100	0.7450	0.7272	1084
neutral	0.6750	0.7050	0.6897	1708
happiness_excited	0.7050	0.6700	0.6871	1636
accuracy			0.7050	5531
macro avg	0.7113	0.7138	0.7122	5531
weighted avg	0.7086	0.7050	0.7054	5531

Figure 5: Performance metrics for Learning Rate Warm-Up and Cosine Annealing with Skip Connections on IEMOCAP dataset

Table 5: Summary of Metrics for IEMOCAP (scripted+improvised, 7 seconds, Focal Loss)

Metric	**Value**
Unweighted Average Recall (UAR)	71.38%
Weighted Accuracy (WA)	70.50%
F1-score (Macro)	71.22%

Overall, this modification improved performance while keeping the model size nearly same.

2 Conclusion

In this work, we introduced targeted modifications to enhance the LIGHT-SERNET model's efficiency and performance in Speech Emotion Recognition, focusing on accuracy, stability, and maintaining a lightweight design for resource-limited applications. By incorporating attention mechanisms, depthwise separable convolutions, skip connections, and a combined learning rate schedule with cosine annealing, we achieved refined feature learning, improved gradient flow, and smoother training.