Efficient Facial Feature Learning with Wide Ensemble-based Convolutional Neural Networks

- ## Aim:

Perform FER using the Ensembles with Shared Representations (ESRs) network and compare the results with single network and traditional ensemble networks.

- ## Methodology:

- Prior work on FER employed the idea of single NNs or an ensemble of different NNs. This paper utilizes ESRs, proposed by the same authors in an earlier published paper.
- ESRs consist of two building blocks, 1) the base of the network – an array of common conv layers (shared layers) for low and mid level feature learning and, 2) these informative features are then shared with independent conv branches that constitute the ensemble. From here, each branch can learn distinctive features while competing for a common resource – shared layers.
- This ensures that the ESRs can leverage the power of numerous neural network branches (or Ensembling) and obtain a reduction in computational load, number of parameters and training time through its shared layers at the base of the network. The shared layers at the beginning offer better generalization as well (due to the reduced number of parameters when compared to a traditional ensemble (TE) network) resulting in state of the art performance on chosen datasets.

- Competitive training is done using the combined loss function given by the summation of loss functions of each branch:

$$L_{esr} = \sum_b \sum_i L[P(f(x_i) = y_i | x_i, \theta_{shared}, \theta_b), y_i]$$

  where, b = branch index,

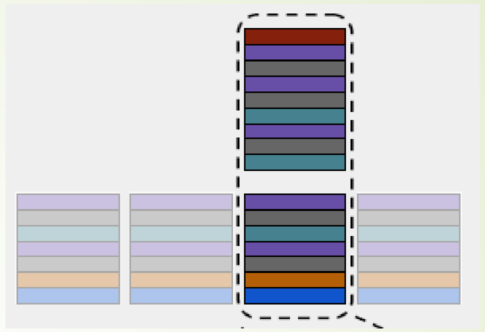
  ($x_{i,}$ y_i) = random training samples, Θ's = parameters of shared and branch layers.

- New conv branches are added in sequence while training. As a result, the shared layers trained earlier on with a single conv branch add a transfer learning effect that guides and accelerates learning as the ensemble grows. The algorithm for training is as given below:

```
Algorithm 1: Training ESRs.
initialize the shared layers with $\theta_{shared}$
for $b$ to maximum ensemble size do
    initialize the convolutional branch $B_b$ with $\theta_b$
    add the branch $B_b$ to the network $ESR$
    sample a subset $D'$ from a training set $D$
    foreach mini-batch $(x_i, y_i) \sim D'$ do
        perform the forward phase
        initialize the combined loss function $L_{esr}$ to 0.0
        foreach existing branch $B_{b'}$ in $ESR$ do
            compute the loss $L_{b'}$ with respect to $B_{b'}$
            add $L_{b'}$ to $L_{esr}$
        end
        perform the backward phase
        optimize $ESR$
    end
end
```

As can be observed, for each iteration on the 2nd line, a new conv branch is added to the network. A sample from the training set is then taken and forward and backward propagation is performed as usual to optimize the combined loss. After the first epoch, much of the parameters of the shared layers have already been learned and this provides a transfer learning effect for the coming epochs and branch training.

- Following is the ESR architecture with 4 conv branches:



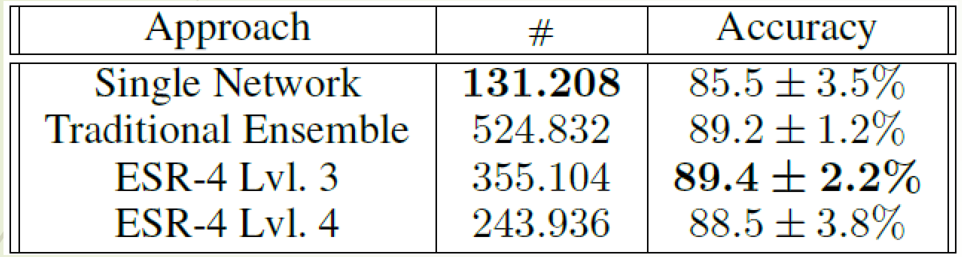
- The datasets used for training are CK+ (small in the lab dataset) and AffectNet, FER+ (larger in the wild datasets to check the networks generalization and scalability).
- 2 baselines – 1) A single 5 layer CNN with BatchNorm, Max and Avg pooling and FC layer

  2) A traditional ensemble of 4 of the above single networks

  are used for comparing the ESRs performance.
- Cross validation is used for the purpose of training on CK+ while training samples of 5000 are prepared using the AffectNet and FER+ datasets with equi representation from each emotion for the purpose of tackling the problem of under representation of certain emotions in these datasets. After adding a new conv branch to the ESR the shared layers and already trained branches continue training on additional data using: (1) the same initial learning rate (fixed lr.: 0.1), (2) a smaller learning rate (varied lr.: lr_{sl} = 0.1 and lr_{tb} = 0.02), or (3) not training at all (frozen layers: lr_{sl}=lr_{tb}=0.0)

Results:

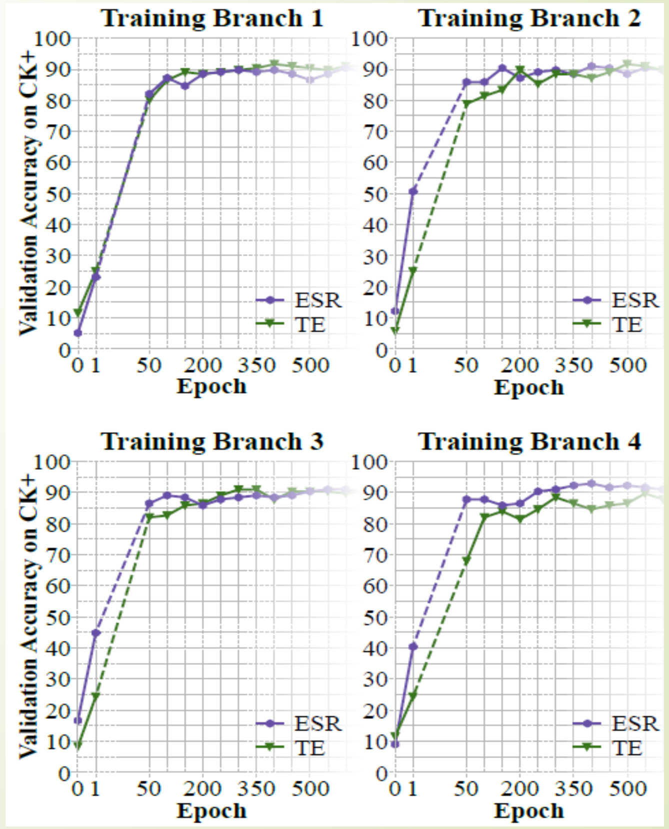
- Results on CK+ and number of parameters:



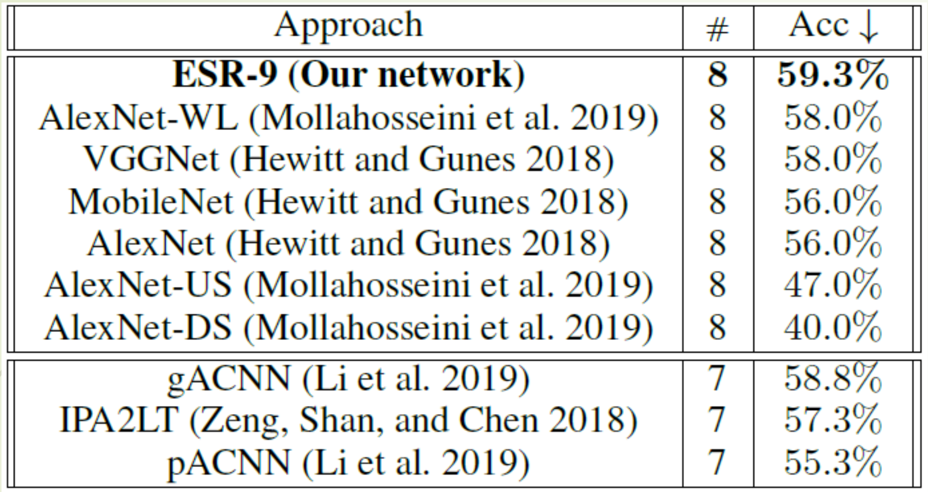
Approach	#	Accuracy
Single Network	**131.208**	$85.5 \pm 3.5\%$
Traditional Ensemble	524.832	$89.2 \pm 1.2\%$
ESR-4 Lvl. 3	355.104	$\mathbf{89.4 \pm 2.2\%}$
ESR-4 Lvl. 4	243.936	$88.5 \pm 3.8\%$

This demonstrates that the ESR – 4 performs the best. Though the accuracy is only marginally increased from the TE network, a massive improvement is seen in the number of parameters and thus in the training time and computational load.

- The figure on the right demonstrates the transfer learning ability of ESRs. With successive increment in the conv branches, the ESR reaches a higher val acc. than TE in the same 50 epochs as the previous branches have been trained already.

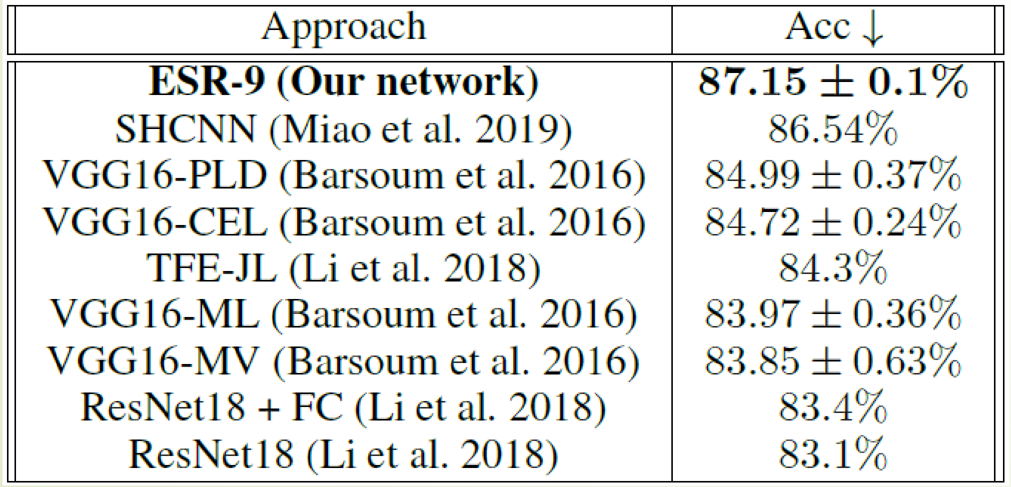


- The results obtained on the AffectNet are provided on the next slide. State of the art results (59.3%) obtained for this dataset which demonstrates that the residual generalization error was reduced owing to the reduced parameters through shared layers:



Approach	#	Acc ↓
ESR-9 (Our network)	**8**	**59.3%**
AlexNet-WL (Mollahosseini et al. 2019)	8	58.0%
VGGNet (Hewitt and Gunes 2018)	8	58.0%
MobileNet (Hewitt and Gunes 2018)	8	56.0%
AlexNet (Hewitt and Gunes 2018)	8	56.0%
AlexNet-US (Mollahosseini et al. 2019)	8	47.0%
AlexNet-DS (Mollahosseini et al. 2019)	8	40.0%
gACNN (Li et al. 2019)	7	58.8%
IPA2LT (Zeng, Shan, and Chen 2018)	7	57.3%
pACNN (Li et al. 2019)	7	55.3%

- The ESR trained on the AffectNet was directly tested on the FER+ dataset. Finetuning on this dataset was done thereafter. Again, state of the art results (87.15%) obtained on this dataset:



Approach	Acc ↓
ESR-9 (Our network)	**$87.15 \pm 0.1\%$**
SHCNN (Miao et al. 2019)	86.54%
VGG16-PLD (Barsoum et al. 2016)	$84.99 \pm 0.37\%$
VGG16-CEL (Barsoum et al. 2016)	$84.72 \pm 0.24\%$
TFE-JL (Li et al. 2018)	84.3%
VGG16-ML (Barsoum et al. 2016)	$83.97 \pm 0.36\%$
VGG16-MV (Barsoum et al. 2016)	$83.85 \pm 0.63\%$
ResNet18 + FC (Li et al. 2018)	83.4%
ResNet18 (Li et al. 2018)	83.1%

- In conclusion, the ESR model helps generalize better to the datasets and also drastically reduces computational load. The model is therefore scalable for large scale datasets and produces state of the art results. However, the problem of catastrophic forgetting needs to be kept in mind while model training. In order to ensure that the model's previously trained layers do not forget previous learned parameters on retraining with new data, the learning rates for these layers need to be very carefully set.

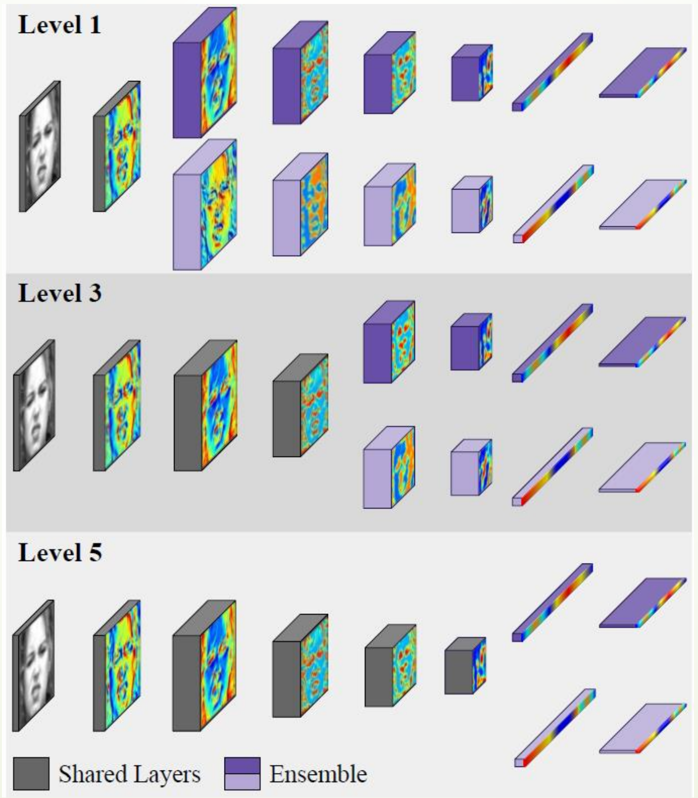


Fig. A summary of the ESR architecture with branching at different depths in the network