Facial Expression Recognition

Abbas Mammadov, Kaleb Mesfin Asfaw, Zahra Bayramli, Tivan Varghese George

TABLE OF CONTENTS

01

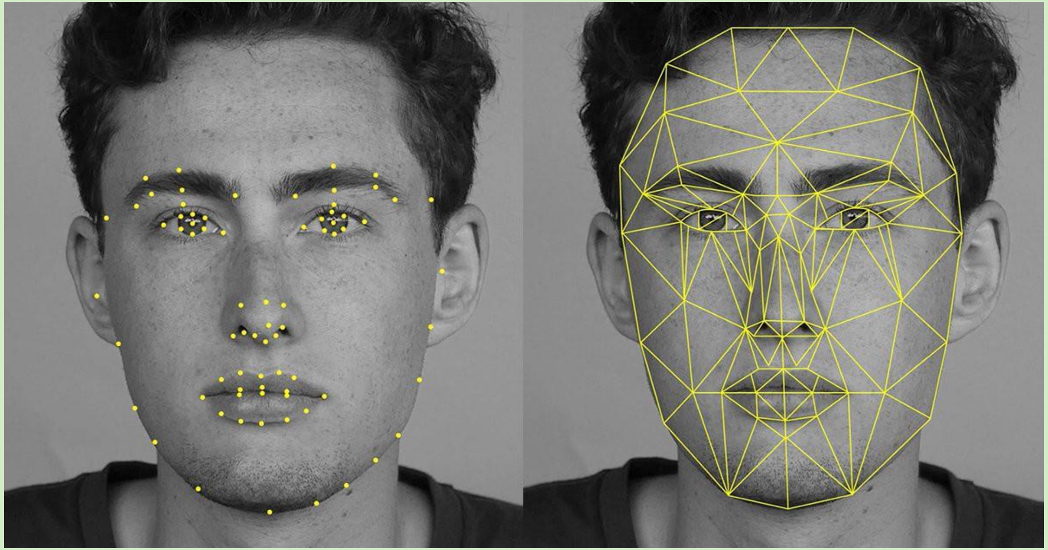
Introduction

Motivation & Baseline Dataset (FER2013)

Introduction and Motivation

- Even though facial expression recognition models already exist, there is room to improve when it comes to real world settings.

- Facial expression recognition can potentially be very useful in the medical, psychiatric, legal, and leisure industries.



Baseline Dataset

FER2013[1]

- Collected by Google image search API
- ~36k grayscale images, resized to 48x48 pixel
- 7 facial expressions – angry, disgust, fear, happy, sad, surprise, neutral
- Train, validation, test sets ratio – 80:10:10

Figure: Example images from FER2013 dataset

02

Related Works

Approach of similar research and their shortcomings

Related Paper

In-Kyu Choi, Ha-eun Ahn and Jisang Yoo
“Facial expression classification using deep CNN” [2]:

- A deep CNN for recognition of 6 expressions
- 10 different datasets
- Cross validation
- Short execution time and good performance (accuracy 93.95 %)

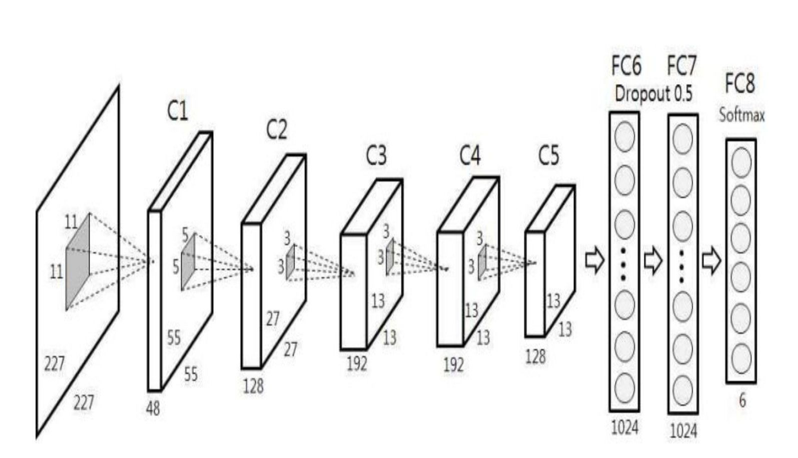


Figure: Proposed CNN architecture

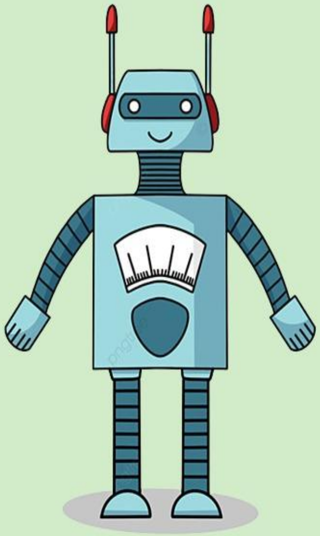
Related Paper

Pramerdorfer, C., Kampel, M. "FER using CNNs: state of the art"[3]

- Ensemble of 8 CNNs from 6 different recent papers
- Test accuracy of 75.2% on FER 2013 dataset

Z. Zhang, P. Luo, C.-C. Loy, and X. Tang, "Learning Social Relation Traits from Face Images":[4]

- 75.1% of accuracy through fusing data from multiple sources



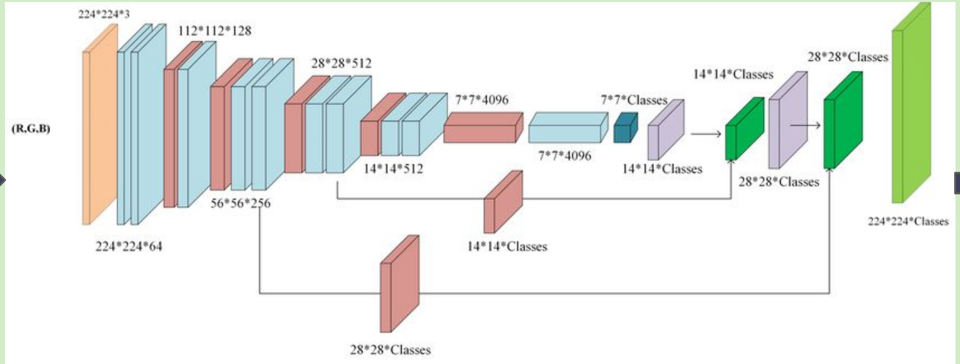
AI

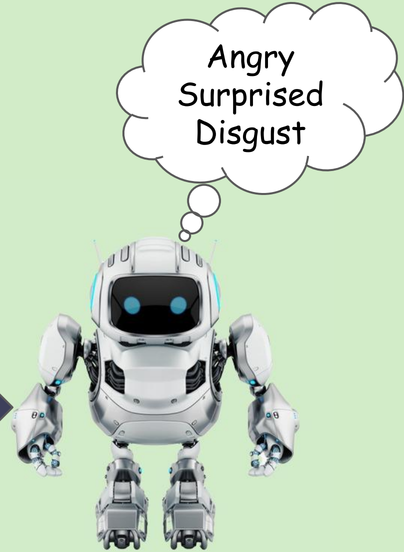
03

Goals & Challenges

Main Objectives of “Facial Expression Recognition”

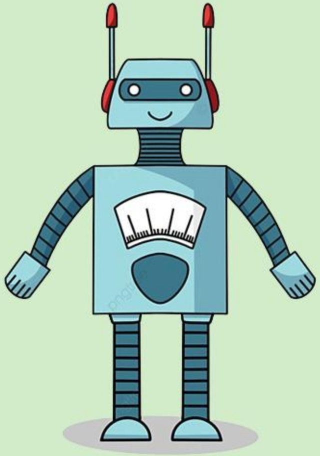
SMART AI





SMART AI should classify the emotions most accurately

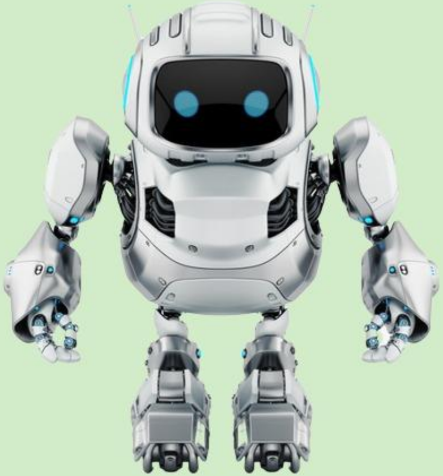
The Breakthrough



Secret Recipe

- A new benchmark dataset
- Open-source Pretrained Language Models
- Transfer learning

AI → SMART AI



The Challenges

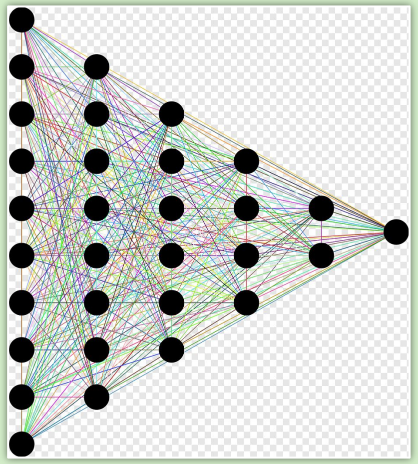
Datasets

Same distribution

Preprocessing each

Unbalanced dataset

Confusion matrix depicting that FER is unbalanced

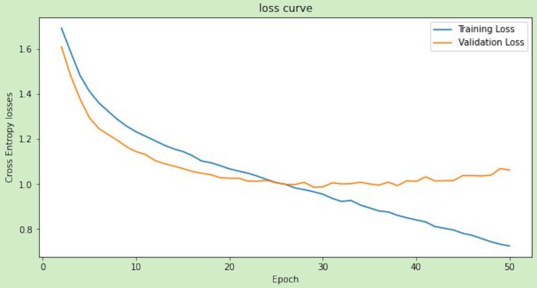


Very Deep

Neural Networks

Computational resources

Potential over-fittings



Resnet-50 trained only on FER2013

Solution

04

Diving into benchmarked techniques, methodologies, and *proposed models*

ADDRESSING OVERFITTING - Part 1

Action on our Dataset -> Data Augmentation and Auxiliary Datasets

Augmentation Techniques

- Horizontal Flipping (randomly)
- Rotation between 10 degrees (randomly)
- Translating within 10 pixels of range (using Random Affine)

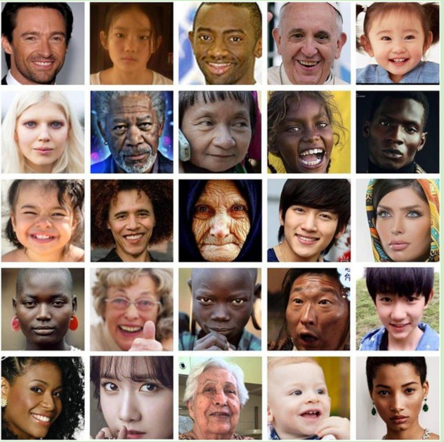
Using *Torchvision Transforms* Library

N.B We also have used the ***Transforms*** Library to convert the images into tensors of ***NORMALIZED PIXELS*** (Using ToTensor function).

Auxiliary Datasets

FER2013

+



AffectNet[5]

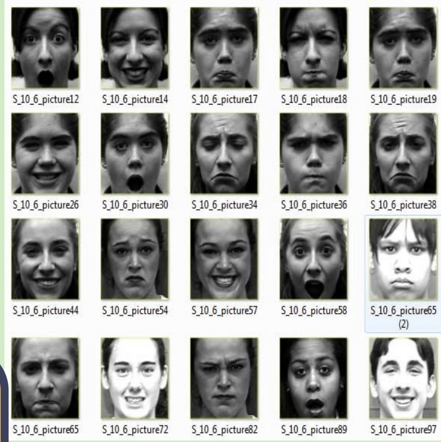
Including only some sample (~30k images) from main AffectNet dataset, which was composed of 0.4 million colored images.

Data cleaning of incorrectly labeled data.

CK+[6]

Including ~1000 balance labeled images from CK+ dataset, which is well fitted for Facial Expression Recognition.

Merging it to the baseline train set to enlarge the samples for training.



ADDRESSING OVERFITTING - Part 2

Actions on Model -> Hyperparameter Tuning (Regularization)

-> **80:10:10 Split**

- **Training**
- **Cross-validation**
- **Test**

Regularization

- *L2 Weight decay*
- *Dropout*
- *Batch Normalization*

Along with SGD (Nesterov) and Adam optimizers

Results

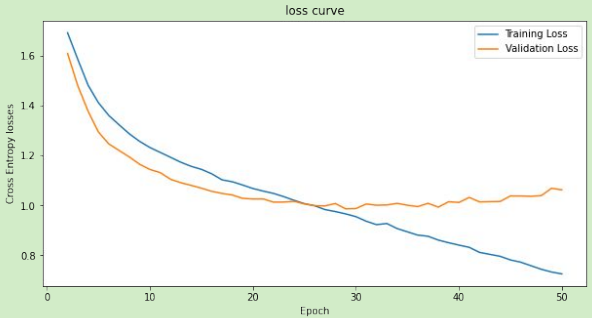
05

Experimental analysis & results

Only with FER2013

-> 80:10:10 Split

- Training
- Cross-validation
- Test



Training and validation loss curves

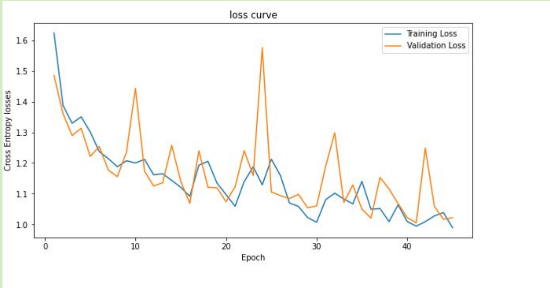
Cross validation Metrics

	RESNET50	RESNET50(T)	VGG19
Accuracy (%)	61.4	64.4	62.6
Loss (NNL)	1.15	1.06	1.02

FER2013 + CKplus + Affectnet

-> 80:10:10 Split

- Training
- Cross-validation
- Test



Training and validation loss curves

Cross validation Metrics

	RESNET50	RESNET50(T)	EFFICIENTNET
Accuracy (%)	61.4	62.2	67
Loss (NNL)	1.04	1.02	1.36

What did we learn?

06

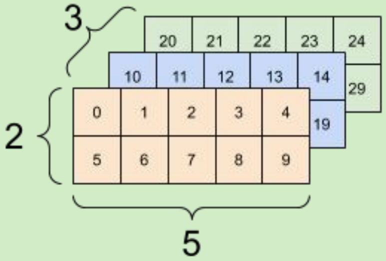
Data Preprocessing

Obtaining data

- Convert csv files into images, and feed the image folders into the data loader using torchvision's inbuilt functions like *ImageFolder.*

Cleaning data

- Check whether our data contains the correct labels.
- If not, make use of pseudo labeled csv files to classify them to their right category.



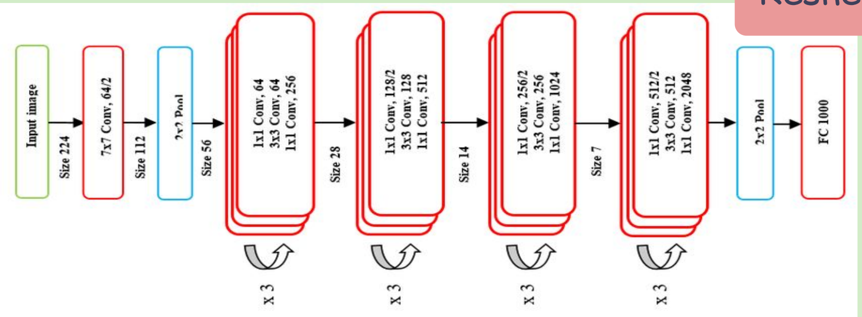
Model Building, and Fine Tuning

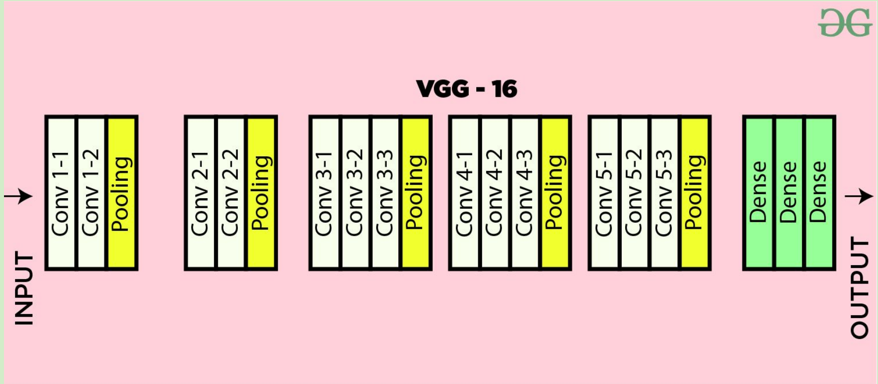
- Examined and compared the performance of a couple of CNN architectures.

- In theory, we only know how the models undergo the forward propagation and the backward propagation process

- However, in this project, we have gained a hands-on experience on fine tuning each of these models we have worked on.

Resnet50

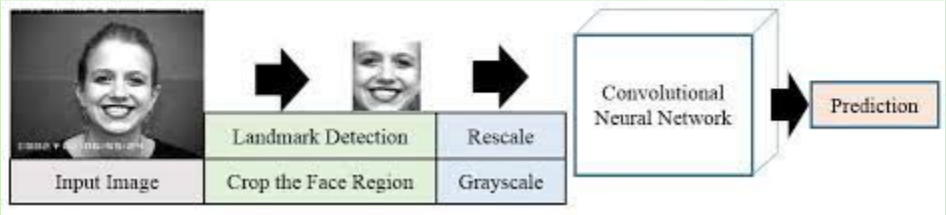




- How to effectively classify our data into train, validation and test sets.
- Able to diagnose overfitting
- Error Analysis Methods
- Saving and loading Models in PyTorch

Model Performance

- We have learnt that a model should be supplied with larger amount of data to prevent overfitting.
- In order to learn from all emotions, it is highly preferable to obtain a dataset with balanced proportion of labels as much as possible
- At last we have indeed saw that ResNet and EfficientNet Architectures are well suited for this task



Summary of Our workflow

References

1. https://www.kaggle.com/datasets/msambare/fer2013
2. https://www.koreascience.or.kr/article/JAKO201809253681042.pdf?fbclid=IwAR3gnwoUoE_arj0gKlfp8uM4Ih92eIwMcpChEw9CKooZHwHT8v9OI19dV5E
3. Pramerdorfer, C., Kampel, M.: Facial expression recognition using convolutional neural networks: state of the art. Preprint arXiv:1612.02903v1, 2016.
4. Z. Zhang, P. Luo, C.-C. Loy, and X. Tang, "Learning Social Relation Traits from Face Images," in Proc. IEEE Int. Conference on Computer Vision (ICCV), 2015, pp. 3631–3639.
5. https://paperswithcode.com/dataset/affectnet
6. https://www.kaggle.com/datasets/shawon10/ckplus

Thank you!