Facial Emotion Recognition using AI

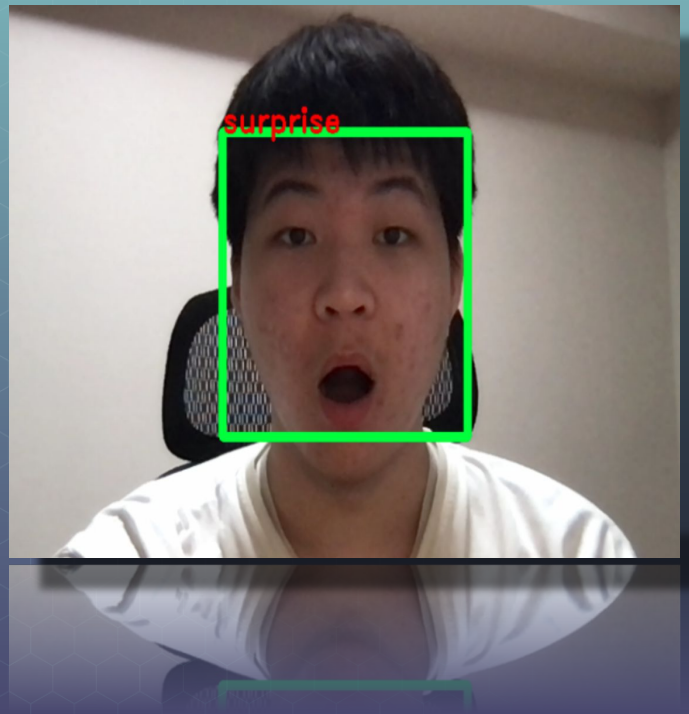
Juneha Hwang, Canadian Academy

Juneha (Harold)

I am a grade 10 student attending Canadian Academy, Kobe. I have lived in South Korea for 3 years before coming to CA in grade 9. I speak Korean, English, and a little bit of Japanese.

Facial Emotion Recognition Program

A facial emotion recognition (FER) program uses AI to classify a person's emotion into six categories: "happy", "neutral", "sad", "angry", "fear", "surprise". It uses a camera to first detect the face. Then, the program analyses the face to find the person's emotion.



Why this program?

I decided to make a FER program because I was interested in AI. In order to build programs using AI, I needed to code. Although I had experience building programs by coding in different languages, AI requires extensive knowledge of math and algorithms. I thought that building this program would really challenge my math and coding abilities.

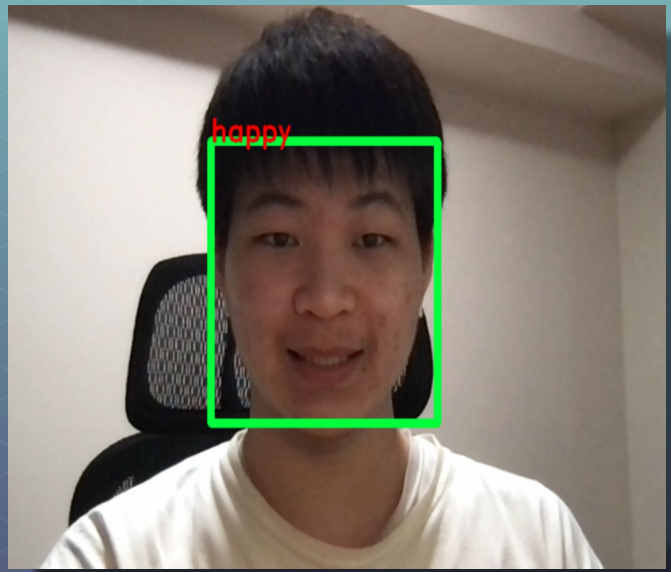


TABLE OF CONTENTS

Criterion A

Investigating

The Purpose of this Project

The Goal

Enhancing my coding capabilities through developing an AI program.

Personal Connection

When I began coding, I always wanted to pursue a route in AI.

Challenge

AI programs are challenging because they are based on math and algorithms not taught in high school.

Photo by Ilya Pavlov on Unsplash

What Did I Already Know?

Coding

I knew programming languages like C/C++, Python, and Java. Python is a common language used for AI.

Online Courses

I took online courses on AI provided by institutes like Harvard and IBM. These courses were loaded with necessary concepts.

Past Projects

I had experience building projects using code. Although they aren't as advanced as this project, the experience helped me overcome obstacles.

Research Process

Primary Source

I read papers and viewed many projects that were similar to my project. I also managed to interview an expert on my topic for pieces of advice.

Secondary Source

I read books related to Deep Learning and useful articles on AI. These helped me know what I needed to study prior to building my program.

Examples of Research – Bibliography, OPVL

Your Name: Juneha

Personal Project Research and Bibliography Worksheet

Please use this worksheet to help you record your research and keep track of the information and resources that you are gathering. For more information on evaluating resources, follow this OPVL Chart

Primary Sources (2-4)

A primary source is an immediate first-hand account of a topic from a person who experienced it
Examples: interviews, performances, pieces of artwork, class lectures, diaries, photographs

How to cite:

- an interview
- a performance or lecture or piece of music
- a piece of artwork
- a photograph (digital) (print)
- just about anything

My Primary Sources

Item	APA Citation	Notes
A paper on reinforcement learning and computer vision	Bernstein, A. & Burnaev, E. (2018). *Reinforcement learning in computer vision*. Retrieved November 17, 2020, from https://www.researchgate.net/publication/324558061_Reinforcement_learning_in_computer_vision	Not really related to FER but is a good reference.
Research paper on FER using deep learning	Mellouk, W. & Handouzi, W. (2020, August 06). *Facial emotion recognition using deep learning: Review and insights*. Retrieved November 18, 2020, from https://www.sciencedirect.com/science/article/pii/S1877050920 31801	This describes the technique and process used behind FER with an image dataset.
A Compact Embedding for Facial Expression Similarity (Research paper)	Agarwala, A., & Vemulapalli, R. (2018). *A Compact Embedding for Facial Expression Similarity*.	This describes how the computer actually predicts people's emotions.
Interview about Facial Emotion Recognition (FER) with an expert	Park, G. (2020, November 29). *Interview about Facial Emotion Recognition (FER) with an expert* [Online interview].	I gained specific feedback on the datasets available for my project. The expert had experience with a similar project so he had lots of advice.
Emotion detection Github repository	atulapra/Emotion-detection. (2020). Retrieved 24 November 2020, from https://github.com/atulapra/Emotion-detection	Has code for making an FER program. The README file is insightful.

Personal Project OPVL Worksheet

Refer to this image for OPVL details

Primary Sources (2-4)

Resource (link or title)	O: Origin	P: Purpose	V: Value	L: Limitation
Interview about Facial Emotion Recognition (FER) with an expert	An expert who has experience with emotion recognition projects and computer vision. He has made a project using brain waves to analyze emotions. He is 32 years old and is South Korean.	This is an opinion from the expert. He conveyed tips on how to choose an effective dataset for my level. He also recommended which dataset to use and gave general pieces of advice when doing a project on FER.	Improved my plan for the project because he gave useful advice on meeting the deadline. He gave some specific directions I could head to next.	This is the opinion of only one person. Thus, cannot determine if other people agree. He didn't reveal the results of his previous projects. He only told the process of it.
Emotion detection Github repository	Created by a user called atulapra. He is a Deep Learning Research Engineer at IIT. The repository is licensed by MIT, which adds credibility. Also, the repository was starred by over 400 people, meaning that many people bookmarked the project.	To share his FER project on the Internet with developers around the world. To keep his project and code on a cloud service and not local.	Provides code and a README that specifies the accuracy and results of the project. Can gain general insight of my project of facial emotion recognition.	Doesn't give specific notes on how the code works. Also, doesn't have a lot of information about the technique used behind the project. Did not show anywhere why he worked on the project

Secondary Sources (4-10)

Resource (link or title)	O: Origin	P: Purpose	V: Value	L: Limitation
A paper on reinforcement learning and computer vision	It is a conference paper published by Alexander Bernstein and Evgeny	To provide a description of reinforcement learning and how it can be used in	Gives some insight about computer vision. Facial emotion recognition uses	Doesn't provide any direct information about FER. Could only be used as a

*OPVL is edited

Primary Sources – Pieces of Code, Interview

104 lines (94 sloc) 2.96 KB Raw Blame

```python
import numpy as np
import pandas as pd
from PIL import Image
from tqdm import tqdm
import os

# convert string to integer
def atoi(s):
    n = 0
    for i in s:
        n = n*10 + ord(i) - ord("0")
    return n

# making folders
outer_names = ['test','train']
inner_names = ['angry', 'disgusted', 'fearful', 'h
os.makedirs('data', exist_ok=True)
for outer_name in outer_names:
    os.makedirs(os.path.join('data',outer_name), e
    for inner_name in inner_names:
        os.makedirs(os.path.join('data',outer_name

# to keep count of each category
angry = 0
disgusted = 0
fearful = 0
happy = 0
sad = 0
surprised = 0
neutral = 0
angry_test = 0
disgusted_test = 0
fearful_test = 0
happy_test = 0
sad_test = 0
surprised_test = 0
neutral_test = 0
```

Pieces of Code on the Internet

Search or jump to... / Pull requests Issues Marketplace Explore

atulapra / Emotion-detection

Watch 19 Unstar 415 Fork 213

Code Issues 1 Pull requests Actions Projects Wiki Security Insights

master 1 branch 0 tags Go to file Add file Code

atulapra Update requirements.txt 8ea28a5 on Dec 21, 2020 55 commits

imgs	Changed folder names	11 months ago
src	Changed plotting code to make it compatible with current versions	9 months ago
LICENSE	Add LICENSE file	16 months ago
README.md	Updated README	6 months ago
requirements.txt	Update requirements.txt	2 months ago

README.md

Emotion detection using deep learning

Introduction

This project aims to classify the emotion on a person's face into one of **seven categories**, using deep convolutional neural networks. The model is trained on the **FER-2013** dataset which was published on International Conference on Machine Learning (ICML). This dataset consists of 35887 grayscale, 48x48 sized face images with **seven emotions** - angry, disgusted, fearful, happy, neutral, sad and surprised.

About

Real-time Facial Emotion Detection using deep learning

opencv computer-vision deep-learning tflearn opencv-python haar-cascade emotion-detection emotion-recognition

Readme

MIT License

Releases

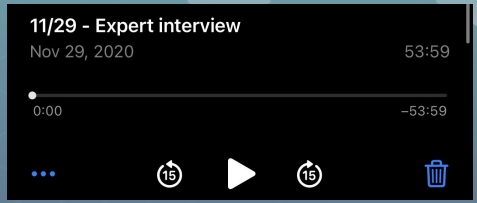
No releases published

Packages

No packages published

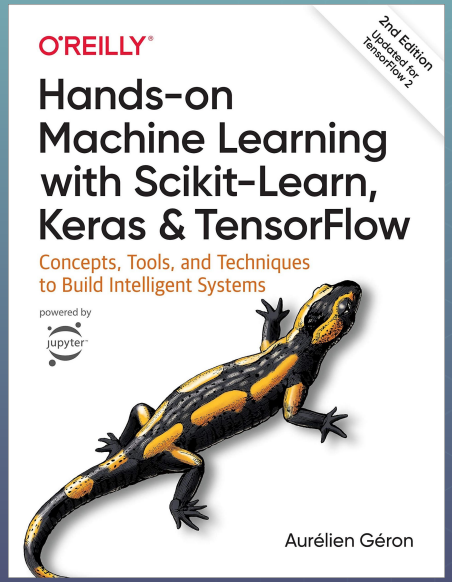
Contributors 3

atulapra Atul Balaji

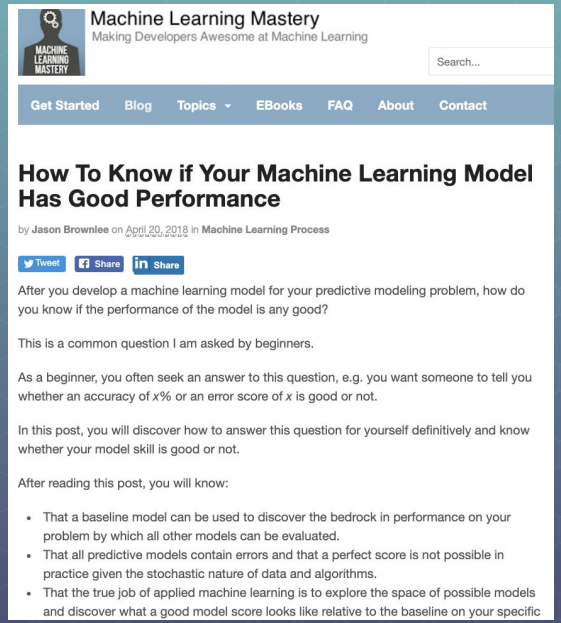


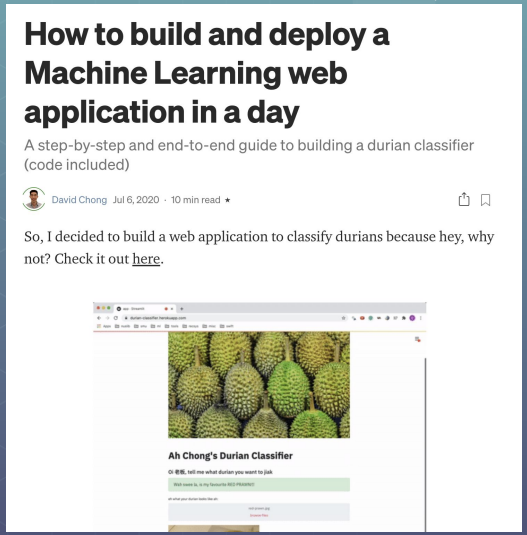
Interview with Expert

Secondary Sources – Books, Articles



Book





Articles

What I learnt from Researching

- A specific type of AI technology for my product
- An overview and necessary steps needed for my project
- I will need to keep on researching while I make my product.

ATL – Research Skills

Collecting useful and relevant data is important because there is not a bunch of sources on my topic

I had to make connections and combine information from sources

I identified and incorporated primary and secondary sources

Collect and analyze the different datasets firsthand

Criterion B

Planning

Success Criteria

Data Visualization

I have to know what data I have to train the computer.

Building Model

I have to build the AI program by coding.

Hyperparameter Tuning

Changing some parts of the code can dramatically change the performance.

Accuracy

My program has to accurately predict the emotion.

Deployment

I have to be able to use the program through a camera.

Success Criteria Rubric

My Name: Juneha	My Goal: Building a FER Program using artificial intelligence
My Product or Outcome: FER Program	My Global Context: Scientific & Technological Innovation

Level of Achievement	*Data Visualization*	*Building program through code*	*Hyperparameter Tuning*	*Accuracy*	*Deployment*
Substantial (5-6)	❑ *Visualizes some unprocessed data* ❑ *Visualizes some processed data* ❑ *Visualize model training process* ❑ *Visualize results using some graphs and metrics.*	❑ *The model uses CNN architecture. It has convolution layers, pooling, and fully connected layers.* ❑ *Model is a deep network. Contains more than two layers* ❑ *The model contains one Callback function.*	❑ *Use tuning strategies such as Grid search, Random search.* ❑ *Show more than 2 percent improvement in validation accuracy after tuning.* ❑ *Attempt changing more than two parameters*	❑ *The final accuracy of the model exceeds 70 percent in the validation set* ❑ *The accuracy of the model exceeds 75 percent in the training dataset.*	❑ *Runs trained model live on a computer camera* ❑ *Separate code for running the model in a live camera* ❑ *Upload code on Github or Kaggle*
Rigorous (7-8)	❑ *Visualizes unprocessed data* ❑ *Visualizes processed data* ❑ *Visualize model training process* ❑ *Visualize results using appropriate graphs and metrics*	❑ *The model uses CNN architecture. It has convolution layers, pooling, and fully connected layers.* ❑ *Model is a deep network. Contains more than four layers* ❑ *The model uses recent technology* ❑ *The model contains several Callback functions.*	❑ *Use multiple tuning strategies such as Grid search, Random search, bayesian optimization.* ❑ *Show more than 5 percent improvement in validation accuracy after tuning.* ❑ *Attempt changing more than three parameters*	❑ *The final accuracy of the model exceeds 75 percent in the validation set* ❑ *The accuracy of the model exceeds 80 percent in the training dataset.*	❑ *Runs trained model live on a computer camera* ❑ *Separate code for running the model in a live camera* ❑ *Deploys model on the internet (web page)* ❑ *The web page has professional design features.* ❑ *Upload code on Github and Kaggle*

*Success Criteria Rubric is edited

Action Plan

Steps	Pre-September	September	October	November	December	January	February	March
Investigation								
confirming goal and global context								
collecting primary sources (interviews, onsite research)								
collecting secondary research (reading artilces, blogs, reliable websites)								
confirm product/outcome of project (event, digital product, object ...)								
Planning								
developing success criteria for project (design specifications)								
completing action plan / timeline for project								
making necessary bookings or purchases								
Taking Action								
Learn the necessary skills to build my product								
Finding dataset and building the program.								
confirm product meets success criteria								
refine product								
Deploy/share the product								
Reflecting								
confirming the extent to which the goal was met								
writing Criteria A								
writing Criteria B								
writing Criteria C								
writing Criteria D								
completing report Appendices								
preparing exhibition								
Exhibition								
collecting further feedback								
redrafting report								
submitting final report								

Plan for Building the Program

Learn the necessary skills
By November 7th

Find Dataset
By December 19th

Build the program
By January 16th

Make changes to the program
By January 23rd

Deploy the program
By January 30th

Dataset I used

ATL – Self management, Planning

Self-management

Coding is difficult and I was going to run into a lot of errors. Therefore, I needed to stay resilient and motivated. Coding also requires developing new skills along the way.

Planning

Because the difficulty of the project could differ from the initial choices I made. I had to not only manage my time, but identify clear goals and steps that I could achieve.

Criterion C

Taking Action

Process of making the Program

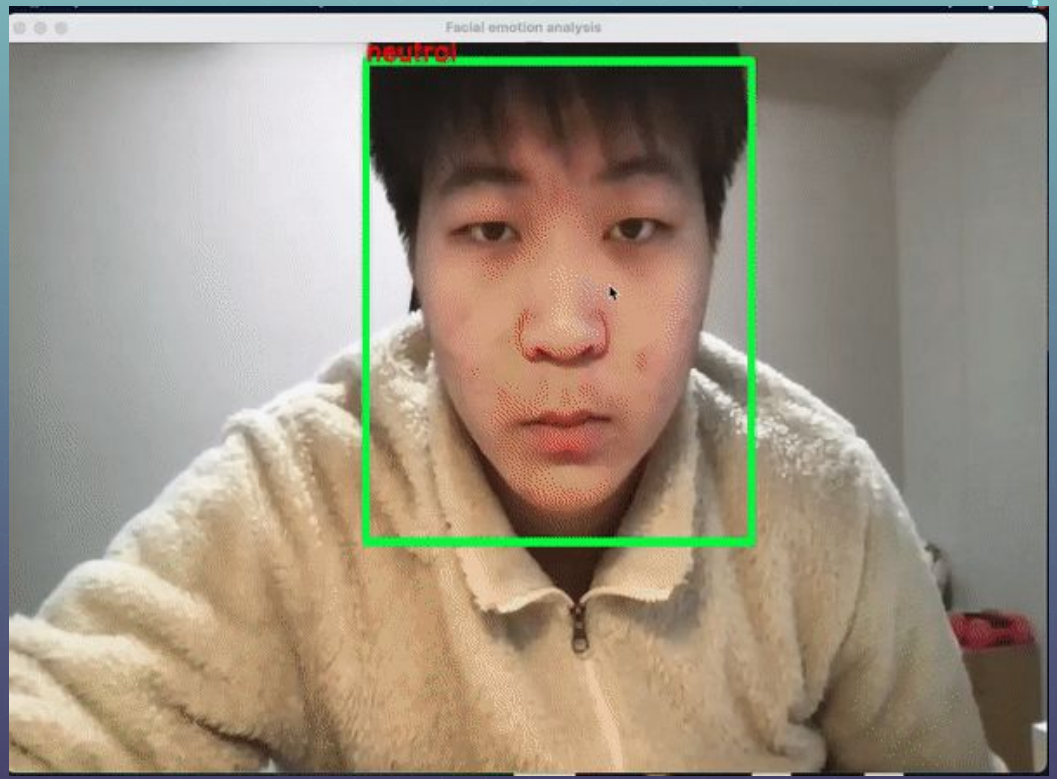
```
class FER_HyperModel(HyperModel):

    def __init__(self, classes):
        self.classes = classes

    def build(self, hp):
        net = Sequential(name='DCNN')

        net.add(
            Conv2D(
                filters = hp.Int('layer1_filter', 8, 64, 8),
                kernel_size=(5,5),
                input_shape=(img_width, img_height, img_depth),
                activation = hp.Choice('layer1_activation', values=['elu', 'swish']),
                padding='same',
                kernel_initializer='he_normal',
                name='conv2d_1'
            )
        )
        net.add(BatchNormalization(name='batchnorm_1'))
        net.add(
            Conv2D(
                filters=hp.Int('layer2_filter', 8, 64, 8),
                kernel_size=(5,5),
                activation=hp.Choice('layer2_activation', values=['elu', 'swish']),
                padding='same',
                kernel_initializer='he_normal',
                name='conv2d_2'
            )
        )
```

Code from my program

```
Epoch 1/100
1010/1010 [==============================] - 13s 13ms/step - loss: 1.8861 - accuracy: 0.2980
acy: 0.4310
Epoch 2/100
1010/1010 [==============================] - 13s 13ms/step - loss: 1.4185 - accuracy: 0.4425
acy: 0.4868
Epoch 3/100
1010/1010 [==============================] - 13s 13ms/step - loss: 1.2914 - accuracy: 0.4964
acy: 0.5430
Epoch 4/100
1010/1010 [==============================] - 13s 13ms/step - loss: 1.2249 - accuracy: 0.5278
acy: 0.5665
Epoch 5/100
1010/1010 [==============================] - 14s 13ms/step - loss: 1.1671 - accuracy: 0.5535
acy: 0.5815
Epoch 6/100
1010/1010 [==============================] - 13s 13ms/step - loss: 1.1308 - accuracy: 0.5673
acy: 0.6080
Epoch 7/100
1010/1010 [==============================] - 13s 13ms/step - loss: 1.0953 - accuracy: 0.5836
acy: 0.5982
Epoch 8/100
1010/1010 [==============================] - 13s 13ms/step - loss: 1.0598 - accuracy: 0.5981
acy: 0.6113
Epoch 9/100
1010/1010 [==============================] - 13s 13ms/step - loss: 1.0353 - accuracy: 0.6107
acy: 0.6197
```

The program is learning through images

Final Product/Outcome



Link to Code:
https://github.com/charrybot/FER-PP

Running the Program Live

What Changes Did I Make?

68% → 70%

Accuracy

20 tunings

Tuning the Code

20 → 28

Blocks of Code

ATL – Thinking, Social, Communication

Social

I kept in contact with the expert I interviewed. He gave me tips along the way with some useful sources.

Thinking

The process of making my project involved thinking skills. Every code contained logical relevance to other parts. Solving error required analyzing

Communication

I used subject specific language and shared my project online. I made inferences to gather useful information.

ATL Snapshots

In [76]:

```
"""model_yaml = model.to_yaml()
with open("model.yaml", "w") as yaml_file:
    yaml_file.write(model_yaml)"""

model.save("model.h5")
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-76-b08e5bd47d92> in <module>
      3     yaml_file.write(model_yaml)"""
      4
----> 5 model.save("model.h5")

NameError: name 'model' is not defined
```

Running into Errors

cross entropy Inbox ×

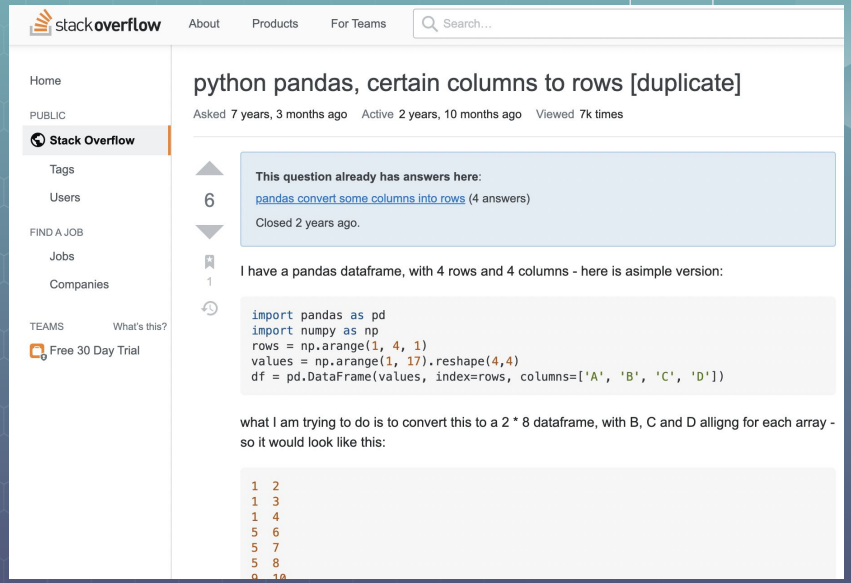
[redacted] to me

https://hyunw.kim/blog/2017/10/26/Cross_Entropy.html

https://chacha95.github.io/2018-11-09-Softmax/

https://cs231n.github.io/neural-networks-case-study/

Communicating with Expert

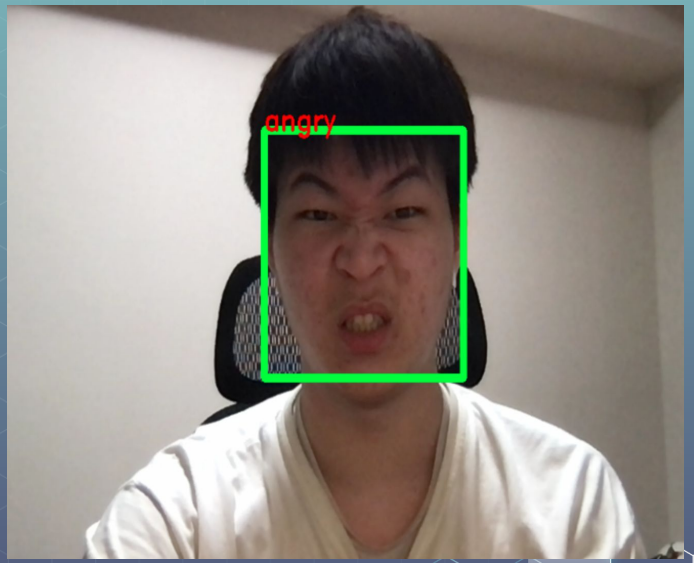


Searching for solutions

Criterion D

Reflecting

My Final Product



Success Criteria Rubric

My Name: Juneha	My Goal: Building a FER Program using artificial intelligence
My Product or Outcome: FER Program	My Global Context: Scientific & Technological Innovation

Level of Achievement	*Data Visualization*	*Building program through code*	*Hyperparameter Tuning*	*Accuracy*	*Deployment*
Substantial (5-6)	❑ *Visualizes some unprocessed data* ❑ *Visualizes some processed data* ❑ *Visualize model training process* ❑ *Visualize results using some graphs and metrics.*	❑ *The model uses CNN architecture. It has convolution layers, pooling, and fully connected layers.* ❑ *Model is a deep network. Contains more than two layers* ❑ *The model contains one Callback function.*	❑ *Use tuning strategies such as Grid search, Random search.* ❑ *Show more than 2 percent improvement in validation accuracy after tuning.* ❑ *Attempt changing more than two parameters*	❑ *The final accuracy of the model exceeds 70 percent in the validation set* ❑ *The accuracy of the model exceeds 75 percent in the training dataset.*	❑ *Runs trained model live on a computer camera* ❑ *Separate code for running the model in a live camera* ❑ *Upload code on Github or Kaggle*
Rigorous (7-8)	❑ *Visualizes unprocessed data* ❑ *Visualizes processed data* ❑ *Visualize model training process* ❑ *Visualize results using appropriate graphs and metrics*	❑ *The model uses CNN architecture. It has convolution layers, pooling, and fully connected layers.* ❑ *Model is a deep network. Contains more than four layers* ❑ *The model uses recent technology* ❑ *The model contains several Callback functions.*	❑ *Use multiple tuning strategies such as Grid search, Random search, bayesian optimization.* ❑ *Show more than 5 percent improvement in validation accuracy after tuning.* ❑ *Attempt changing more than three parameters*	❑ *The final accuracy of the model exceeds 75 percent in the validation set* ❑ *The accuracy of the model exceeds 80 percent in the training dataset.*	❑ *Runs trained model live on a computer camera* ❑ *Separate code for running the model in a live camera* ❑ *Deploys model on the internet (web page)* ❑ *The web page has professional design features.* ❑ *Upload code on Github and Kaggle*

Adequate (3-4)	❑ *Visualizes processed data or unprocessed data* ❑ *Visualize results using uncommon methods*	❑ *The model does not use CNN architecture. It uses a different method.* ❑ *Model is a shallow network. Contains one or two layers* ❑ *The model does not contain Callback functions.*	❑ *Does not use hyperparameter tuning strategies.* ❑ *Show no improvement in validation accuracy after tuning.* ❑ *Attempt changing one parameter*	❑ *The final accuracy of the model exceeds 65 percent in the validation set* ❑ *The accuracy of the model exceeds 70 percent in the training dataset.* 4	❑ *Cannot run trained model live on a computer camera* ❑ *Upload code on Github or Kaggle*
Substantial (5-6)	❑ *Visualizes some unprocessed data* ❑ *Visualizes some processed data* ❑ *Visualize model training process* ❑ *Visualize results using some graphs and metrics.* 6	❑ *The model uses CNN architecture. It has convolution layers, pooling, and fully connected layers.* ❑ *Model is a deep network. Contains more than two layers* ❑ *The model contains one Callback function.*	❑ *Use tuning strategies such as Grid search, Random search.* ❑ *Show more than 2 percent improvement in validation accuracy after tuning.* ❑ *Attempt changing more than two parameters* 5	❑ *The final accuracy of the model exceeds 70 percent in the validation set* ❑ *The accuracy of the model exceeds 75 percent in the training dataset.*	❑ *Runs trained model live on a computer camera* ❑ *Separate code for running the model in a live camera* ❑ *Upload code on Github or Kaggle* 6
Rigorous (7-8)	❑ *Visualizes unprocessed data* ❑ *Visualizes processed data* ❑ *Visualize model training process* ❑ *Visualize results using appropriate graphs and metrics*	❑ *The model uses CNN architecture. It has convolution layers, pooling, and fully connected layers.* ❑ *Model is a deep network. Contains more than four layers* ❑ *The model uses recent technology* ❑ *The model contains several Callback functions.* 7	❑ *Use multiple tuning strategies such as Grid search, Random search, bayesian optimization.* ❑ *Show more than 5 percent improvement in validation accuracy after tuning.* ❑ *Attempt changing more than three parameters*	❑ *The final accuracy of the model exceeds 75 percent in the validation set* ❑ *The accuracy of the model exceeds 80 percent in the training dataset.*	❑ *Runs trained model live on a computer camera* ❑ *Separate code for running the model in a live camera* ❑ *Deploys model on the internet (web page)* ❑ *The web page has professional design features.* ❑ *Upload code on Github and Kaggle*

Photo by K. Mitch Hodge on Unsplash

IB Learner Profiles

Inquirer, Thinker

I have demonstrated being an inquirer and thinker. The main process of my project was writing code. I had to analyse and solve errors. I also had to think of creative and unfamiliar ways to improve my product.

On top of that, I had to keep learning and researching while creating the product because I had to use new methods to make my code work.

Things I Would Change

Dataset I used

Dataset
The dataset had the right difficulty but was not high-quality

More Time
I could have had higher accuracy with more research and analysis of my project

Solving Errors
Took me a lot of time trying to solve the error myself when there was a solution on the Internet.

Things that Went Well

Used a new tool to make my product. Also exposed to other tools.

New Tools

While building my product, I learned how each bit of code works.

Learn Concepts

Final product finds people's emotions in a satisfactory accuracy.

My Code Works

Some Takeaways

1. **Better Planning** - I need to plan according to the time and capabilities I have.
2. **Social Skills** - I think my project involved relatively less social skills than other projects.
3. **Related to Career** - I want to pursue a career in AI. This experience will help me later on.

Questions?

CONTACT

Does anyone have any questions? Email:

23hwajun@canacad.ac.jp
or
junehahwang@gmail.com

CREDITS

Photos are my own. All other photos are from Unsplash

- Presentation template by Slidesgo
- Icons by Flaticon
- Infographics by Freepik
- Images created by Freepik
- Author introduction slide photo created by Freepik
- Text & Image slide photo created by Freepik.com

Thank You