

クラウドソーシングを用いた感情分析

2021/03/23 渡邊真悟



発表の内容

【目的】

- ・人間の顔写真から感情を分類する機械学習モデルの構築

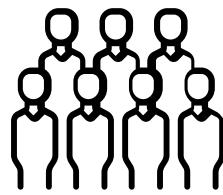
【実験】

- ・AMTにて画像をラベル付けしてもらうタスクを依頼した。
- ・その結果を用いて感情を判定する機械学習モデルを構築した。

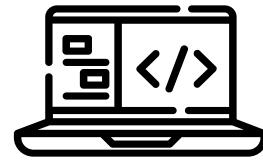
【結果】

- ・テストデータの正解率は約33%にしかならなかった。
- ・人間でも難しい場合での間違いと、特定の不正解クラスを答える場合が目立った。

やったこと



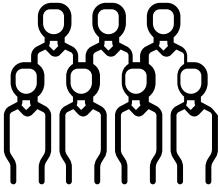
Workerに
タスクを依頼



学習モデルを
作成



テスト



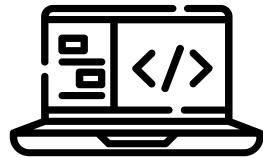
Workerにタスクを依頼

- 顔画像のオープンデータを用いて人間の表情がどんな感情かをタグ付けしてもらうタスクを依頼

What expression is this guy?



- happy
- fear
- angry
- sad



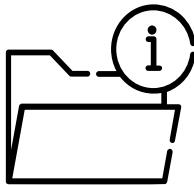
学習モデルを作成

- VGG16をFine-tuning



テスト

- 未知の顔画像から予測させる



使用したデータセットについて

- ・ kaggleで提供されている **MMA FACIAL EXPRESSION** を使用

学習用データをクラウドソーシングでラベル付け
検証用とテスト用のデータはオリジナルのものを使用

- ・ 用いるラベルは以下の7種類



angry



disgust



fear



happy



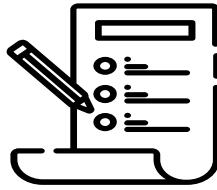
neutral



sad

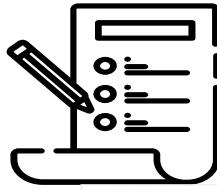


surprise



クラウドソーシングの詳細

- ・ AMTにて実験
- ・ 画像1枚につき \$0.01
- ・ 1画像あたり3人の Worker が解答



タスクの最終設計 UI 1回目

Instructions Shortcuts Choose the correct category



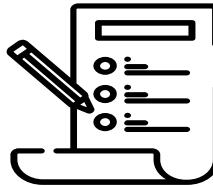
Select an option

angry	1
disgust	2
fear	3
happy	4
neutral	5
sad	6
surprise	7

   
z o m f

Submit

Showing Task 1 of 350 [Next HIT](#)



タスクの最終設計 UI 2回目

正解ラベルをヒントとして表示

Others have determined that this person's emotions are **fear**.

Instructions

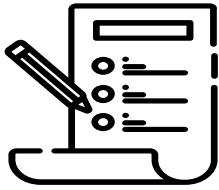
Shortcuts

Choose the correct category



Select an option

angry	1
disgust	2
fear	3
happy	4
neutral	5
sad	6
surprise	7



タスクの最終設計 Instruction 2回目

Look at the picture and imagine what the expression of the person in the picture would look like. Then, choose the best category you thought.

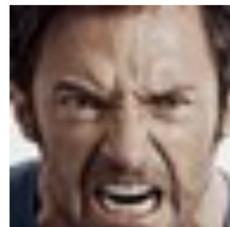
Other people's predictions are written this way as a hint.

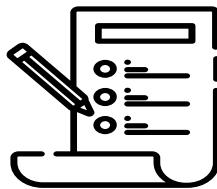
'Others have determined that this person's emotions are angry'

Therefore, if you are not sure whether your prediction is true or not, please refer to the predictions of others listed above.

Here're Just some of examples below.

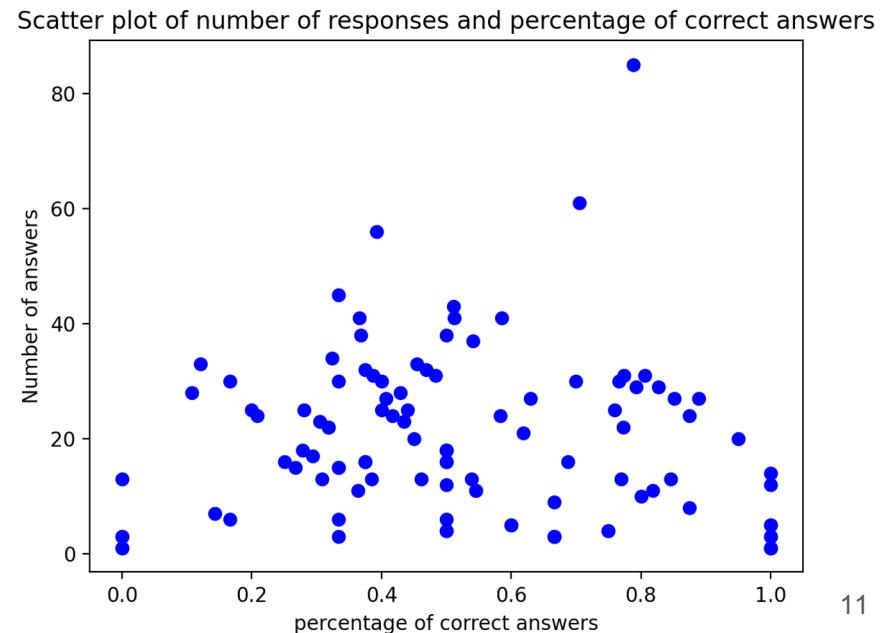
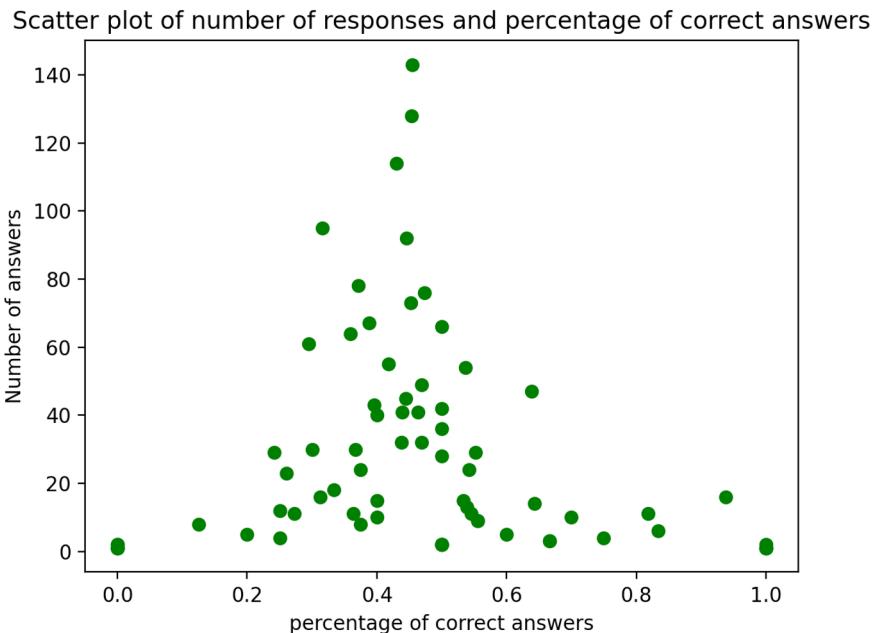
- angry:

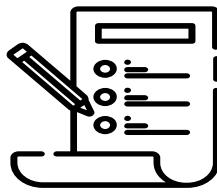




AMTの結果について ワーカーの特徴

緑が1回目 青が2回目 (ヒントありのとき)



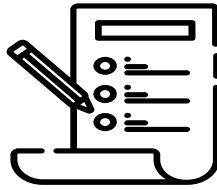


AMTの結果について オリジナルの混同行列

緑が1回目 青が2回目 (ヒントありのとき)

正解クラス												
W o r k e r の 解 答	angry	127	28	64	2	14	19	25	127	0.45519713	0.42333333	0.43868739
	disgust	14	43	17	0	15	15	15	43	0.36134454	0.14333333	0.2052506
	fear	11	18	31	1	7	16	21	31	0.2952381	0.10333333	0.15308642
	happy	28	34	30	263	33	28	52	263	0.56196581	0.87666667	0.68489583
	neutral	51	92	51	19	164	61	31	164	0.34968017	0.54666667	0.42652796
	sad	57	74	83	6	58	156	18	156	0.34513274	0.52	0.41489362
	surprise	12	11	24	9	9	5	138	138	0.66346154	0.46	0.54330709

正解クラス												
W o r k e r の 解 答	angry	130	27	42	2	13	25	19	130	0.50387597	0.43333333	0.46594982
	disgust	22	90	16	1	10	16	13	90	0.53571429	0.3	0.38461539
	fear	19	18	96	3	18	22	36	96	0.45283019	0.32	0.375
	happy	19	32	29	257	34	22	43	257	0.58944954	0.85666667	0.69836957
	neutral	52	71	41	22	177	45	22	177	0.41162791	0.59	0.48493151
	sad	48	57	59	6	43	169	8	169	0.43333333	0.56333333	0.48985507
	surprise	10	5	17	9	5	1	159	159	0.77184466	0.53	0.6284585

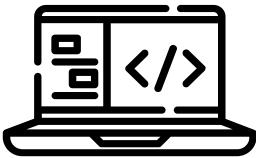


クリーニング手法について

- Dawid-Skeneの手法を用いた
- 青が実験2回目のオリジナル オレンジがクリーニング後

Workerの解答	正解クラス								正解数	再現率	適合率	F値	正解率
	angry	disgust	fear	happy	neutral	sad	surprise						
angry	130	27	42	2	13	25	19	130	0.50387597	0.43333333	0.46594982	0.51333333	
disgust	22	90	16	1	10	16	13	90	0.53571429	0.3	0.38461539		
fear	19	18	96	3	18	22	36	96	0.45283019	0.32	0.375		
happy	19	32	29	257	34	22	43	257	0.58944954	0.85666667	0.69836957		
neutral	52	71	41	22	177	45	22	177	0.41162791	0.59	0.48493151		
sad	48	57	59	6	43	169	8	169	0.43333333	0.56333333	0.48985507		
surprise	10	5	17	9	5	1	159	159	0.77184466	0.53	0.6284585		

Workerの解答	正解クラス								正解数	再現率	適合率	F値	正解率
	angry	disgust	fear	happy	neutral	sad	surprise						
angry	46	9	16	1	3	8	4	46	0.52873563	0.46	0.49197861	0.57	
disgust	8	40	4	0	2	7	4	40	0.61538462	0.4	0.48484848		
fear	7	7	37	0	6	7	10	37	0.5	0.37	0.42528736		
happy	3	10	8	92	13	7	9	92	0.64788732	0.92	0.76033058		
neutral	15	17	11	2	62	14	6	62	0.48818898	0.62	0.54625551		
sad	14	16	18	1	10	57	2	57	0.48305085	0.57	0.52293578		
surprise	7	1	6	4	4	0	65	65	0.74712644	0.65	0.69518717		



機械学習の手法について

VGG16をFine-tuningする

使用したライブラリなど

```
import os
from keras.applications.vgg16 import VGG16
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential, Model
from keras.layers import Input, Activation, Dropout, Flatten, Dense
from keras.preprocessing.image import ImageDataGenerator
from keras import optimizers
import numpy as np

os.environ["KMP_DUPLICATE_LIB_OK"]='TRUE'
os.environ["CUDA_VISIBLE_DEVICES"]="0"

Using TensorFlow backend.
```

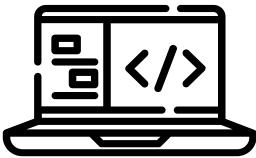
パラメータなど

```
def vgg_model_maker():
    """ VGG16のモデルをFC層以外使用。FC層のみ作成して結合して用意する """
    # VGG16のロード。FC層は不要なので include_top=False
    input_tensor = Input(shape=(img_width, img_height, 3))
    vgg16 = VGG16(include_top=False, weights='imagenet', input_tensor=input_tensor)

    # FC層の作成
    top_model = Sequential()
    top_model.add(Flatten(input_shape=vgg16.output_shape[1:]))
    top_model.add(Dense(256, activation='relu'))
    top_model.add(Dropout(0.5))
    top_model.add(Dense(nb_classes, activation='softmax'))

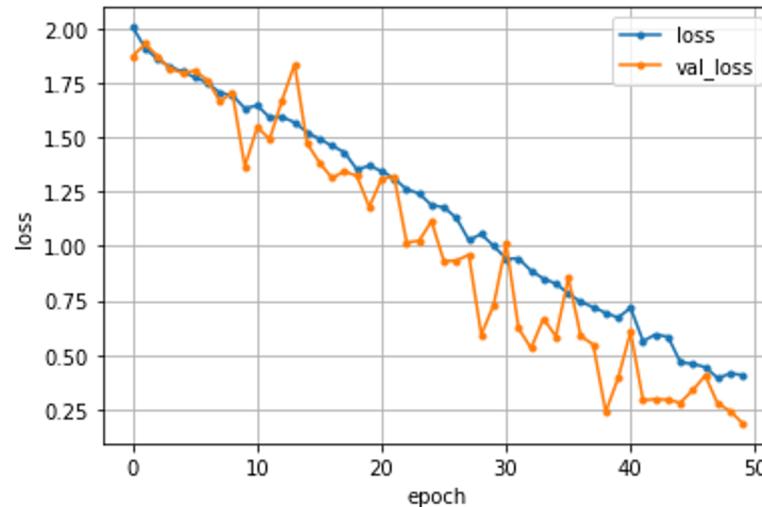
    # VGG16とFC層を結合してモデルを作成
    model = Model(inputs=vgg16.input, outputs=top_model(vgg16.output))

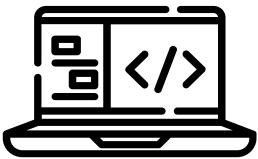
    return model
```



機械学習の結果について

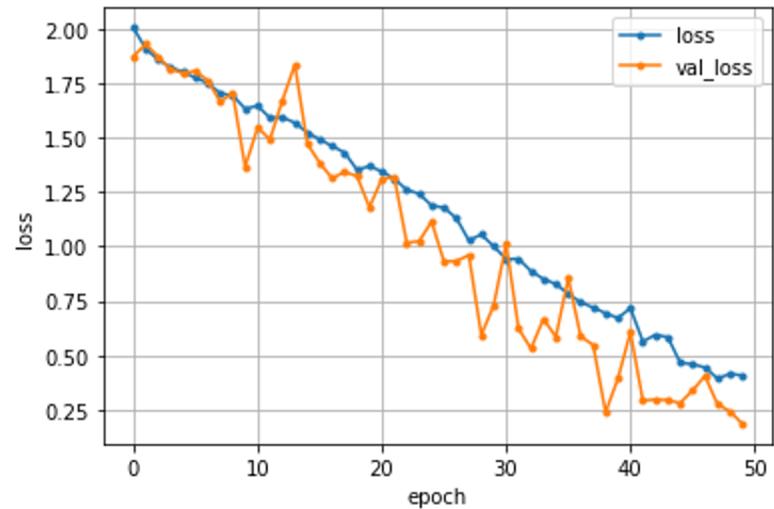
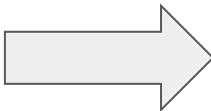
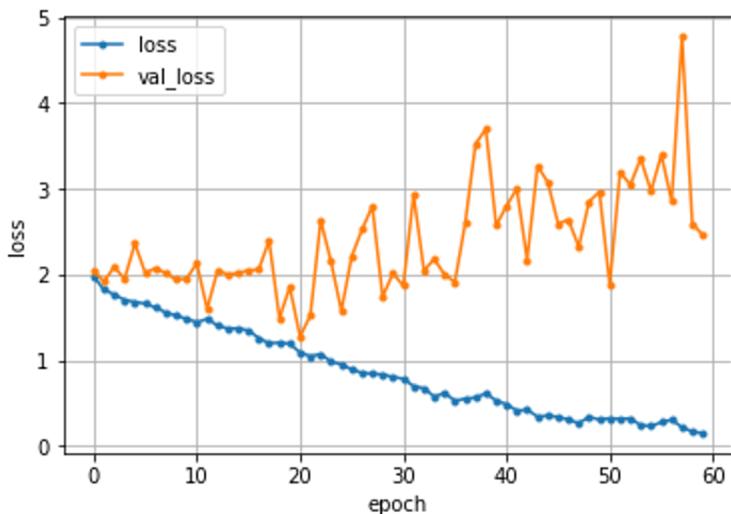
- 各クラス**300枚**になるまで画像を水増しした
- 50エポック**回した





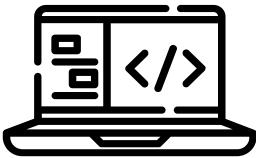
機械学習の結果について

- 画像を水増ししたことで **val_loss** も減っている



中間発表時

今回

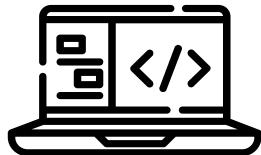


テストの結果について

- 混同行列は以下のようになつた。
- neutral**と予測する結果が多くなつてしまつた。
- 正解率自体は中間発表時よりも良くなつた。 (25% → 33%)

予測結果

	正解クラス												
	angry	disgust	fear	happy	neutral	sad	surprise	正解数	再現率	適合率	F値	正解率	
angry	3	0	1	0	3	0	3	3	0.3	0.6	0.4	0.32857143	
disgust	1	3	0	2	4	0	0	3	0.3	0.42857143	0.35294118		
fear	0	1	3	0	3	0	3	3	0.3	0.5	0.375		
happy	1	1	0	1	5	1	1	1	0.1	0.14285714	0.11764706		
neutral	0	0	0	2	6	0	2	6	0.6	0.21428571	0.31578947		
sad	0	2	1	1	5	1	0	1	0.1	0.5	0.16666667		
surprise	0	0	1	1	2	0	6	6	0.6	0.4	0.48		

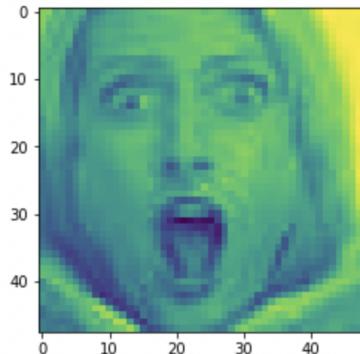


テストの結果について

うまくいかなかった例

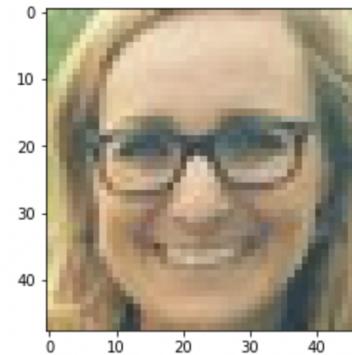
1. 人間でも難しそうなケース

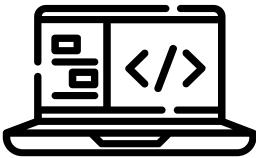
```
=====
./data2/test/fear_00.jpg
[('surprise', 0.9601741), ('fear', 0.037037387)]
```



2. neutralと答えていたりのケース

```
=====
./data2/test/happy_02.jpg
[('neutral', 0.54501784), ('surprise', 0.21758255)]
```



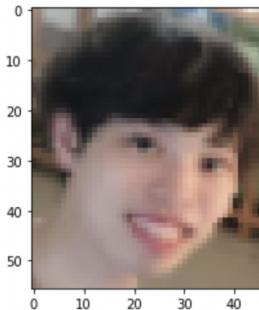


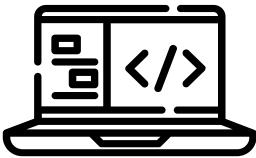
テストの結果について

- ・自分の写真でも試してみた
→ 結果は **neutral** だった



```
./data2/test/me.png
[('neutral', 0.7764495), ('disgust', 0.121140845)]
```





テストの結果について 考察

- **neutral** のラベルは不正解のラベルのばらつきが大きかった

W o r k e r の 解 答	正解クラス								正解数	再現率	適合率	F値	正解率
	angry	disgust	fear	happy	neutral	sad	surprise						
	angry	46	9	16	1	3	8	4	46	0.52873563	0.46	0.49197861	0.57
	disgust	8	40	4	0	2	7	4	40	0.61538462	0.4	0.48484848	
	fear	7	7	37	0	6	7	10	37	0.5	0.37	0.42528736	
	happy	3	10	8	92	13	7	9	92	0.64788732	0.92	0.76033058	
	neutral	15	17	11	2	62	14	6	62	0.48818898	0.62	0.54625551	
	sad	14	16	18	1	10	57	2	57	0.48305085	0.57	0.52293578	
surprise	7	1	6	4	4	0	65	65	0.74712644	0.65	0.69518717		