Hand Gesture Classification with Convolution Neural Networks (CNN)

NATHANIEL SOLON PEDERSEN, University of Stavanger, Norway

Hand gestures plays an important role in non-verbal communication between humans. With machine learning we can predict and classify which hand gesture someone makes. In this study we will be experimenting with different CNNs by tuning different parameters. The dataset we are using consists of labeled Grey-scale images [1]. Source code can be found on Github[3]

Additional Key Words and Phrases: Machine Learning, Convolution Neural Network, Image Classification

ACM Reference Format:
Nathaniel Solon Pedersen. 2024. Hand Gesture Classification with Convolution Neural Networks (CNN) . 1, 1 (November 2024), 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Data preprocessing

The raw dataset consists of Grey-Scale images captured under pretty good conditions, but to improve the model performance, we will need to process the images further.

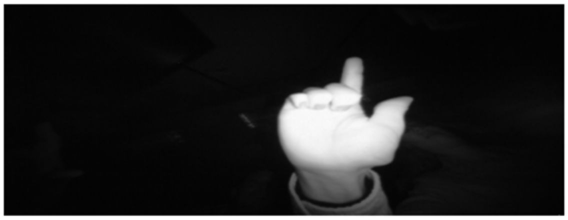


Fig. 1. Image[1] of an 'l' gesture before processing

1.1 Shrinking

Resizing all images into 32x32. This will ensure consistency between all images input into the model. This is also a great way to reduce the amount of data and speed up training time.

1.2 Grey-Scale

We will reduce the dimensionality of the input data from 3 channels (RGB) to a single channel. The colors are not that relevant for hand gesturing and will only add a big computational overhead.

1.3 Blur

A Gaussian blur will be used to remove out noise from the images to improve model accuracy.

1.4 Edge Detection

We are using Canny edge detection to highlight edges. This will just make the gesture generally easier to identify.

1.5 Normalize

Normalization is redundant when using the canny edge detection, but the job of normalization would be to mitigate gradients exploding or vanishing.

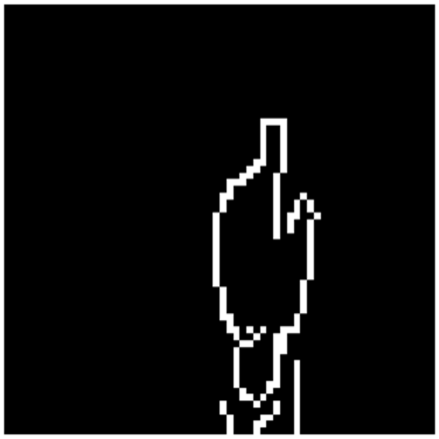


Fig. 2. Image[1] of an 'l' gesture after processing

2 Feature Extraction and Splitting

We will convert all images into numpy arrays, so we can use that as input for tensorflow keras. We will also one hot encode all labels. All data is splitted into training data, validation data, and testing data. The split is 80/10/10. Each data-split will have an equal amount of variety between each label.

3 Models

We will be using 2 different CNN models. Below is the pseudocode for the first model.

```
Input(shape=(64, 64, 1))

Conv2D(32, (3, 3), activation="relu")
MaxPooling2D(pool_size=(2, 2))

Conv2D(64, (3, 3), activation="relu")
MaxPooling2D(pool_size=(2, 2))
```

Author's Contact Information: Nathaniel Solon Pedersen, University of Stavanger, Stavanger, Rogaland, Norway.

ACM XXXX-XXXX/2024/11-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

```
Conv2D(64, (3, 3), activation="relu")
MaxPooling2D((2, 2))

Flatten()
Dense(128, activation="relu")
Dense(10, activation="softmax")

compile(loss="MAE")
```

Below is the pseudocode for the second model

```
Input(shape=(64, 64, 1))

Conv2D(32, (3, 3), activation="relu")
MaxPooling2D(pool_size=(2, 2))

Flatten()
Dense(128, activation="relu")
Dense(10, activation="softmax")

compile(loss="categorical_crossentropy")
```

4 Results

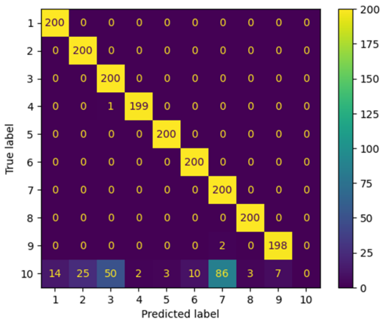
When using the confusion matrix, refer to this table.

Number	Label
1	palm
2	l
3	fist
4	fist moved
5	thumb
6	index
7	ok
8	palm moved
9	c
10	down

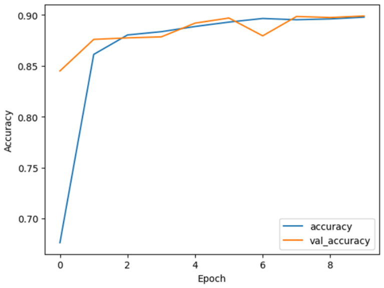
Table 1. Mapping of Numbers to Hand Gesture Labels

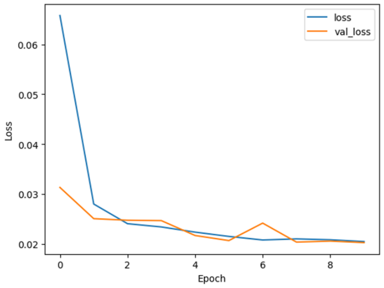
4.1 Experiment 1

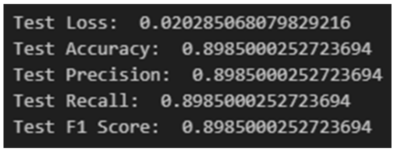
We are using the first model, and training it with an Epoch of 10.



	1	2	3	4	5	6	7	8	9	10
1	200	0	0	0	0	0	0	0	0	0
2	0	200	0	0	0	0	0	0	0	0
3	0	0	200	0	0	0	0	0	0	0
4	0	0	1	199	0	0	0	0	0	0
5	0	0	0	0	200	0	0	0	0	0
6	0	0	0	0	0	200	0	0	0	0
7	0	0	0	0	0	0	200	0	0	0
8	0	0	0	0	0	0	0	200	0	0
9	0	0	0	0	0	0	2	0	198	0
10	14	25	50	2	3	10	86	3	7	0





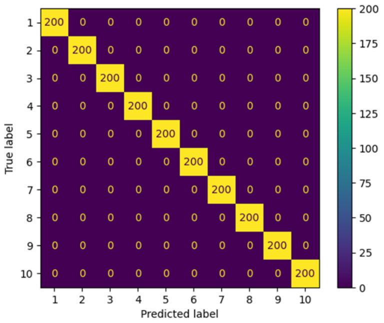


```
Test Loss:  0.020285068079829216
Test Accuracy:  0.8985000252723694
Test Precision:  0.8985000252723694
Test Recall:  0.8985000252723694
Test F1 Score:  0.8985000252723694
```

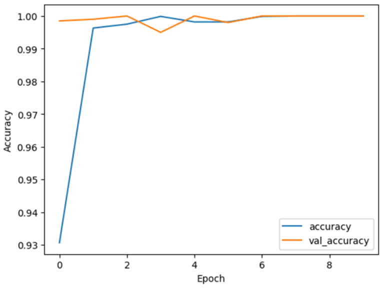
We see that the model is performing with minimal losses, though it struggles with identifying the down gesture.

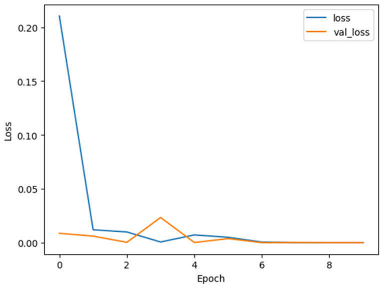
4.2 Experiment 2

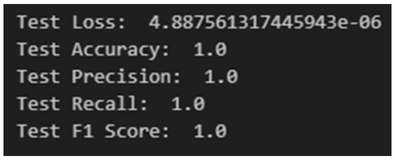
We swapped out from MAE to categorical crossentropy



	1	2	3	4	5	6	7	8	9	10
1	200	0	0	0	0	0	0	0	0	0
2	0	200	0	0	0	0	0	0	0	0
3	0	0	200	0	0	0	0	0	0	0
4	0	0	0	200	0	0	0	0	0	0
5	0	0	0	0	200	0	0	0	0	0
6	0	0	0	0	0	200	0	0	0	0
7	0	0	0	0	0	0	200	0	0	0
8	0	0	0	0	0	0	0	200	0	0
9	0	0	0	0	0	0	0	0	200	0
10	0	0	0	0	0	0	0	0	0	200





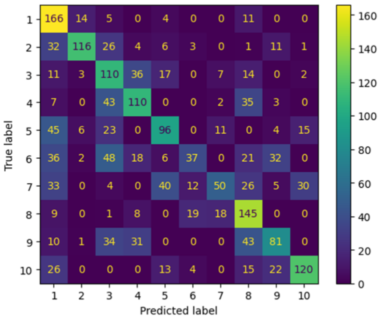


```
Test Loss:  4.887561317445943e-06
Test Accuracy:  1.0
Test Precision:  1.0
Test Recall:  1.0
Test F1 Score:  1.0
```

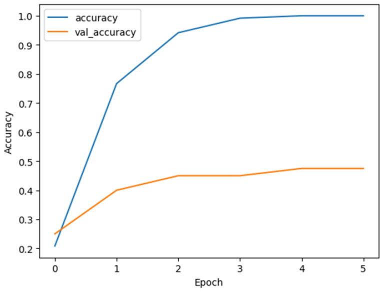
The model correctly predicted all of our tests.

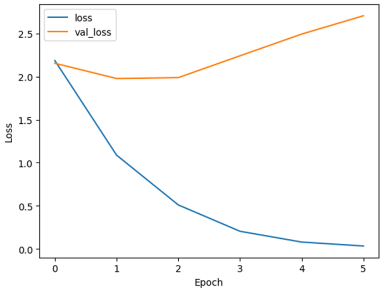
4.3 Experiment 3

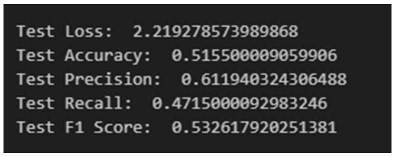
We swapped out to a smaller model. The Advantage of using a smaller model is that it is less computational expensive, but at the sacrifice of it making worse predictions. We also reduced the Epoch to 6, reduced training data down to 100 in length and valuation data down to 40 in length.



True label \ Predicted label	1	2	3	4	5	6	7	8	9	10
1	166	14	5	0	4	0	0	11	0	0
2	32	116	26	4	6	3	0	1	11	1
3	11	3	110	36	17	0	7	14	0	2
4	7	0	43	110	0	0	2	35	3	0
5	45	6	23	0	96	0	11	0	4	15
6	36	2	48	18	6	37	0	21	32	0
7	33	0	4	0	40	12	50	26	5	30
8	9	0	1	8	0	19	18	145	0	0
9	10	1	34	31	0	0	0	43	81	0
10	26	0	0	0	13	4	0	15	22	120





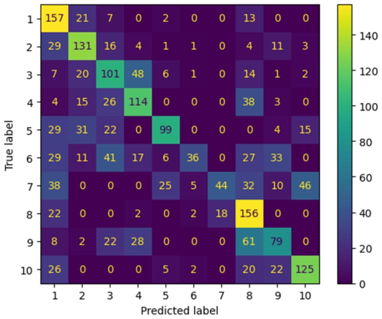


```
Test Loss:  2.219278573989868
Test Accuracy:  0.5155000090599060
Test Precision:  0.611940324306488
Test Recall:  0.4715000092983246
Test F1 Score:  0.532617920251381
```

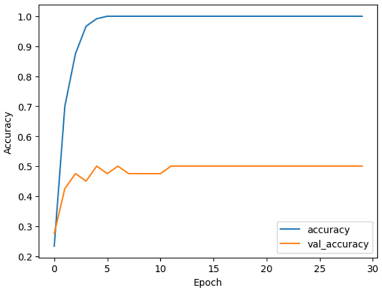
We see that the current model runs worse than our original model.

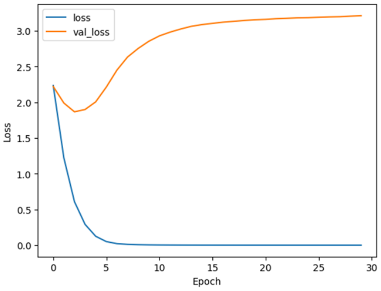
4.4 Experiment 4

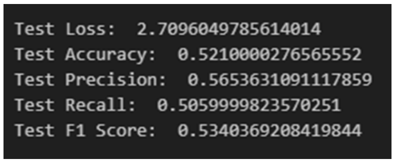
We increased the Epoch to 30.



True label \ Predicted label	1	2	3	4	5	6	7	8	9	10
1	157	21	7	0	2	0	0	13	0	0
2	29	131	16	4	1	1	0	4	11	3
3	7	20	101	48	6	1	0	14	1	2
4	4	15	26	114	0	0	0	38	3	0
5	29	31	22	0	99	0	0	0	4	15
6	29	11	41	17	6	36	0	27	33	0
7	38	0	0	0	25	5	44	32	10	46
8	22	0	0	2	0	2	18	156	0	0
9	8	2	22	28	0	0	0	61	79	0
10	26	0	0	0	5	2	0	20	22	125





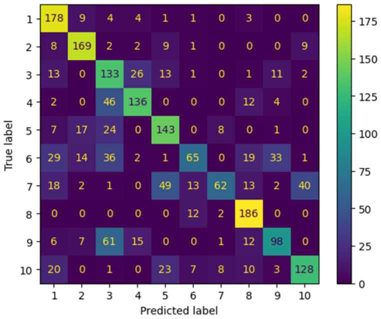


```
Test Loss:  2.7096049785614014
Test Accuracy:  0.5210000276565552
Test Precision:  0.5653631091117859
Test Recall:  0.5059999823570251
Test F1 Score:  0.5340369208419844
```

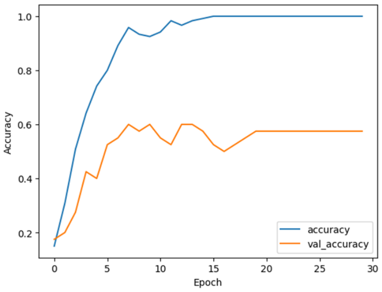
We can observe over fitting, and slight gradient explosions in accuracy when valuating.

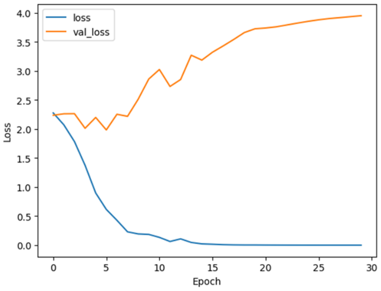
4.5 Experiment 5

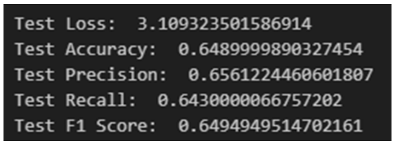
We went back to the previous model.



	1	2	3	4	5	6	7	8	9	10
1	178	9	4	4	1	1	0	3	0	0
2	8	169	2	2	9	1	0	0	0	9
3	13	0	133	26	13	1	0	1	11	2
4	2	0	46	136	0	0	0	12	4	0
5	7	17	24	0	143	0	8	0	1	0
6	29	14	36	2	1	65	0	19	33	1
7	18	2	1	0	49	13	62	13	2	40
8	0	0	0	0	0	12	2	186	0	0
9	6	7	61	15	0	0	1	12	98	0
10	20	0	1	0	23	7	8	10	3	128







```
Test Loss:  3.109323501586914
Test Accuracy:  0.6489999890327454
Test Precision:  0.6561224460601807
Test Recall:  0.6430000066757202
Test F1 Score:  0.6494949514702161
```

We see the same issue with over fitting again, so it means we are lacking data. We also see exploding gradients.

5 Limitations

5.1 Augmentation

When training, we do not take account into rotation nor mirroring, which would be a smart thing to do to have a more robust model.

5.2 Improved datasets

The dataset we are currently using is sampled in infrared and has ideal conditions. In the future, we should use samples with more noise, so we get a more robust model.

6 Acknowledgments

Acknowledgments

To Leap Motion for providing the datasets for this study[1][2]

References

[1] Leap Motion 2016. *Hand Gesture Recognition Database.* Leap Motion, Vienna, Austria. Retrieved November 25, 2024 from https://www.kaggle.com/datasets/gti-upm/leapgestrecog

[2] 2016. *Hand Gesture Recognition using Infrared Imagery Provided by Leap Motion Controller.* https://doi.org/10.1007/978-3-319-48680-2_5

[3] nasope. 2024. *dat305-2024.* github. https://github.com/nasope/dat305-2024