CADDYIAN - Diver Gesture Language Classification

Veronica Peng, Xi Yu, Wenxi Zhao
tpeng24@stanford.edu, isruyuki@stanford.edu, wenxi99z@stanford.edu

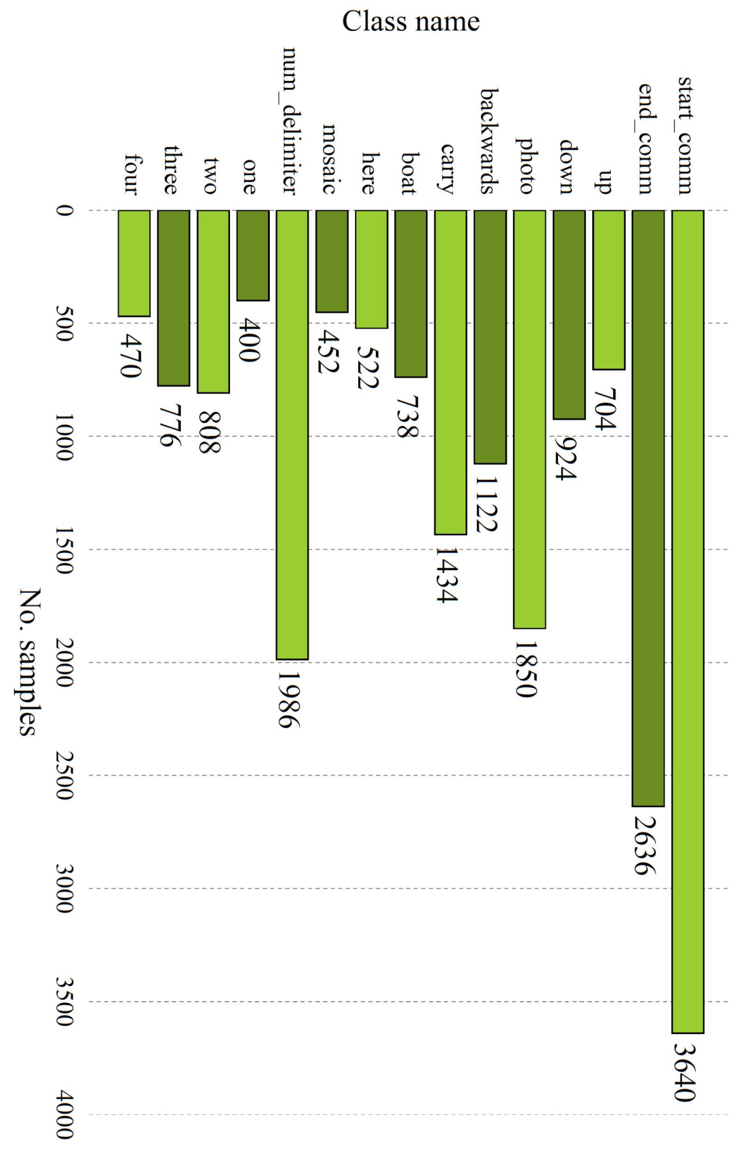
Vedio: https://youtu.be/_OrnuuC4UnM

Predicting

- Autonomous robot companion operated through gesture-based language, CADDYIAN, helps to protect divers in dangerous underwater environments.
- We explore the performance of ***Resnet-18, 50, and 101*** deep learning networks to classify CADDYIAN gestures using the public dataset from the CADDY project [1]. For better performace, we experimented training with both original and ***manually balanced*** datasets, with ***3 resolution levels*** 240×180, 320×240, and 480×360, and with ***categorical cross entropy*** or hinge loss. Our best single model achieved 97.45% test set accuracy with resnet-18 trained on dataset2500, categorical cross entropy, and 240×180 resolution. Our best ensemble model is ***majority voting***, achieving 98.12% test accuracy.
- tSNE analysis indicates that true negative class(with no meaningful gesture) can be easily confused with other classes. Additional error analysis identifies 6 major error factors, and proposes generating more training data and prior hand localization as remedy to further boost performance.

Data & Features

- The CADDY Underwater Stereo-Vision Dataset contains a total of 32858 640*480 RGB images labeled with 16 gesture classes and one true negative class (no gesture).
- Image counts for each class are extremely unbalanced.

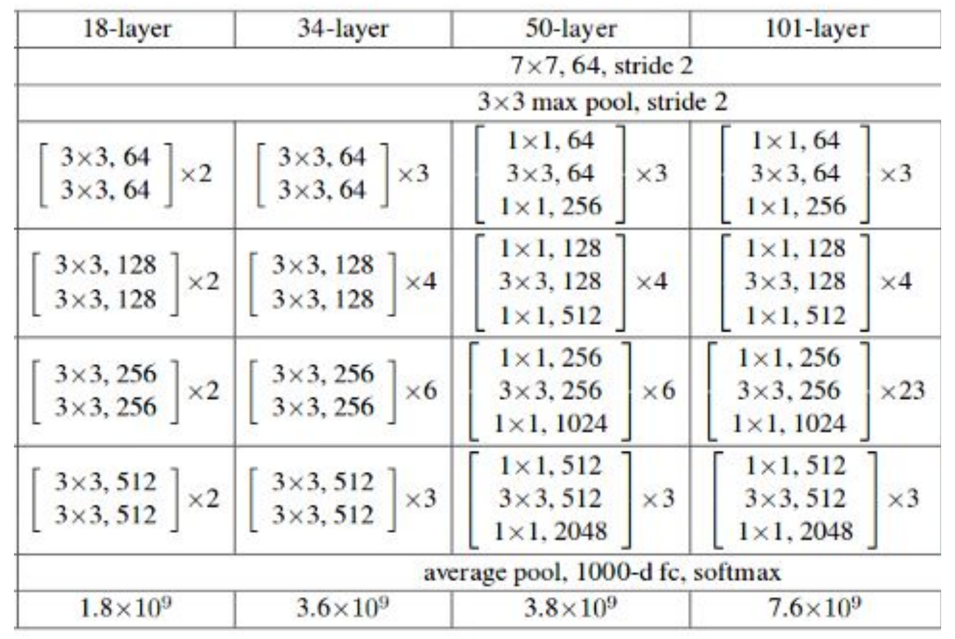


Dataset creation:

- Dataset all-scenarios: 91% train set, 4.5% for both valid & test set. Class weight proportional to the reciprocal of class size.
- Dataset2500: retain all true negative & start_comm images, oversample all other classes with 2500 images. Class weight balanced.

Model

- We used original Resnet-18, Resnet-50, Resnet-101 architecture adapted from CIFAR-10 classification task.
- Best performing resnet-18 has 4 stages. All layers are implemented with 3*3 2D convolution. With proceeding to the next stage, the size of image halve and the number of channels double.



18-layer	34-layer	50-layer	101-layer
7×7, 64, stride 2			
3×3 max pool, stride 2			
$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$
$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$
$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
average pool, 1000-d fc, softmax			
1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9

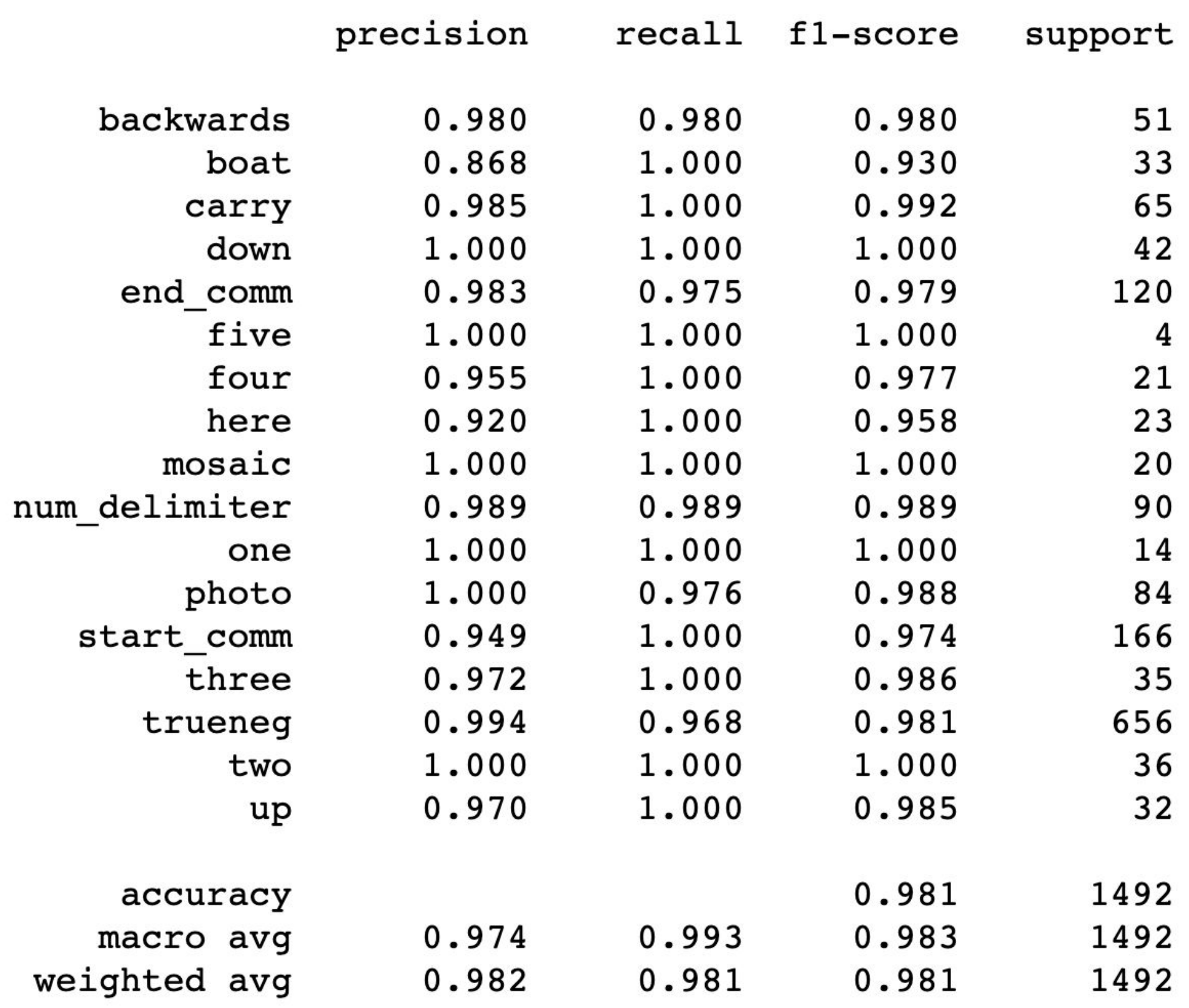
Results

Below are the performance of our single model on the train set (29874 images), dev set (1492 images), and test set (1492 images), and ensemble model on dev set and test set.

Single Models	Train set	Image size	Train set accuracy	Dev set accuracy	Test set accuracy
resnet 101	all-scenarios	240 × 180	88.96%	89.94%	90.41%
resnet 50	all-scenarios	240 × 180	95.74%	95.37%	94.91%
resnet 18	all-scenarios	240 × 180	93.34%	95.10%	95.17%
resnet 50	datset2500	240 × 180	96.45%	96.78%	95.84%
resnet 18	datset2500	240 × 180	96.99%	**97.92%**	97.45%
resnet 18	datset2500	320 × 240	96.96%	97.79%	**97.85%**
resnet 18	datset2500	480 × 360	96.27%	97.39%	96.98%
Ensemble Models				**Dev set accuracy**	**Test set accuracy**
Highest precision				**97.92%**	97.45%
Majority vote				97.72%	**98.12%**

Performance Visualization

Majority Vote (our best ensemble model) has great sub-class performance on test set:

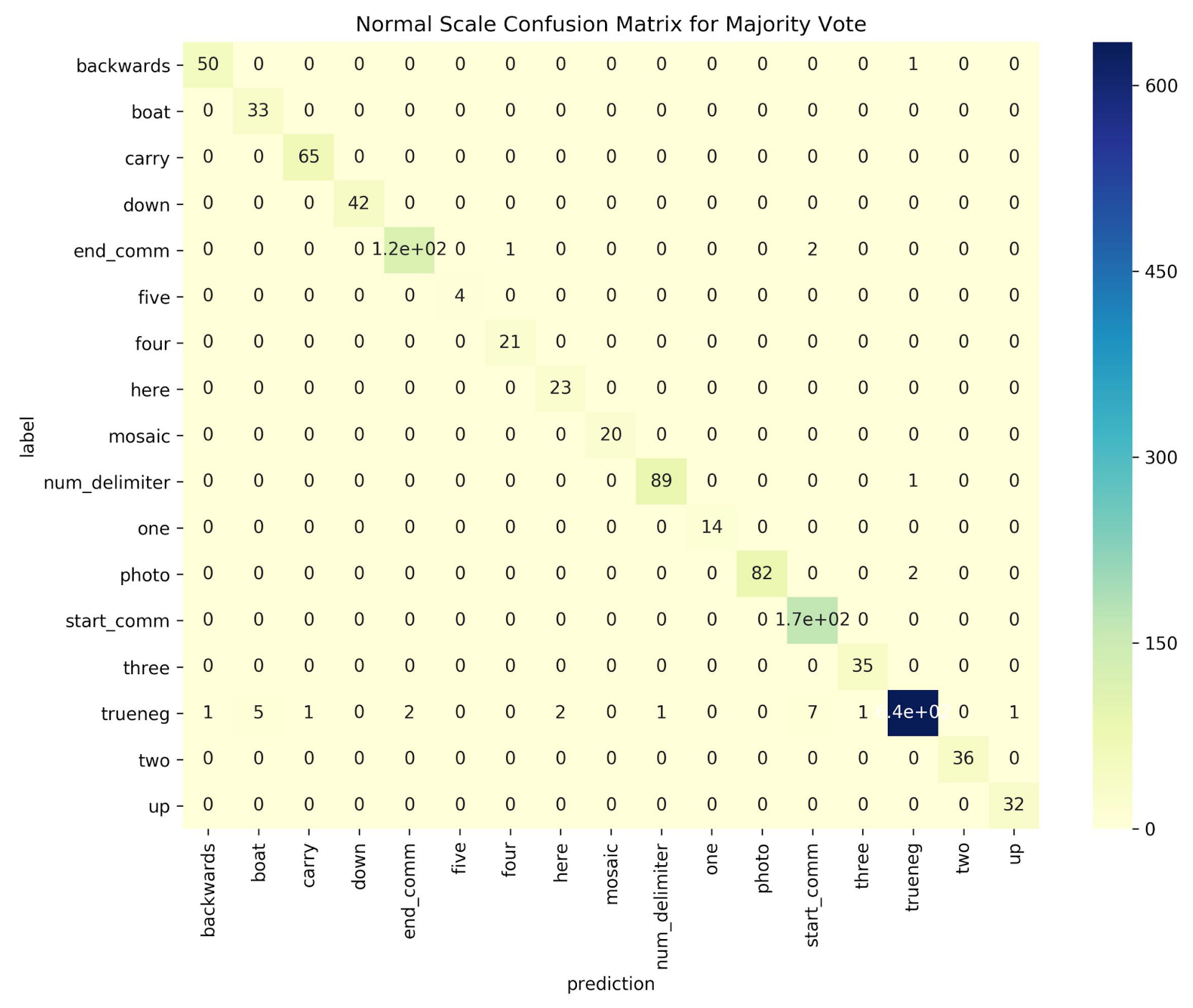


```
              precision    recall  f1-score   support

    backwards      0.980     0.980     0.980        51
         boat      0.868     1.000     0.930        33
        carry      0.985     1.000     0.992        65
         down      1.000     1.000     1.000        42
     end_comm      0.983     0.975     0.979       120
         five      1.000     1.000     1.000         4
         four      0.955     1.000     0.977        21
         here      0.920     1.000     0.958        23
       mosaic      1.000     1.000     1.000        20
num_delimiter      0.989     0.989     0.989        90
          one      1.000     1.000     1.000        14
        photo      1.000     0.976     0.988        84
   start_comm      0.949     1.000     0.974       166
        three      0.972     1.000     0.986        35
      trueneg      0.994     0.968     0.981       656
          two      1.000     1.000     1.000        36
           up      0.970     1.000     0.985        32

     accuracy                          0.981      1492
    macro avg      0.974     0.993     0.983      1492
 weighted avg      0.982     0.981     0.981      1492
```

Sub-class precision, recall and f1-score

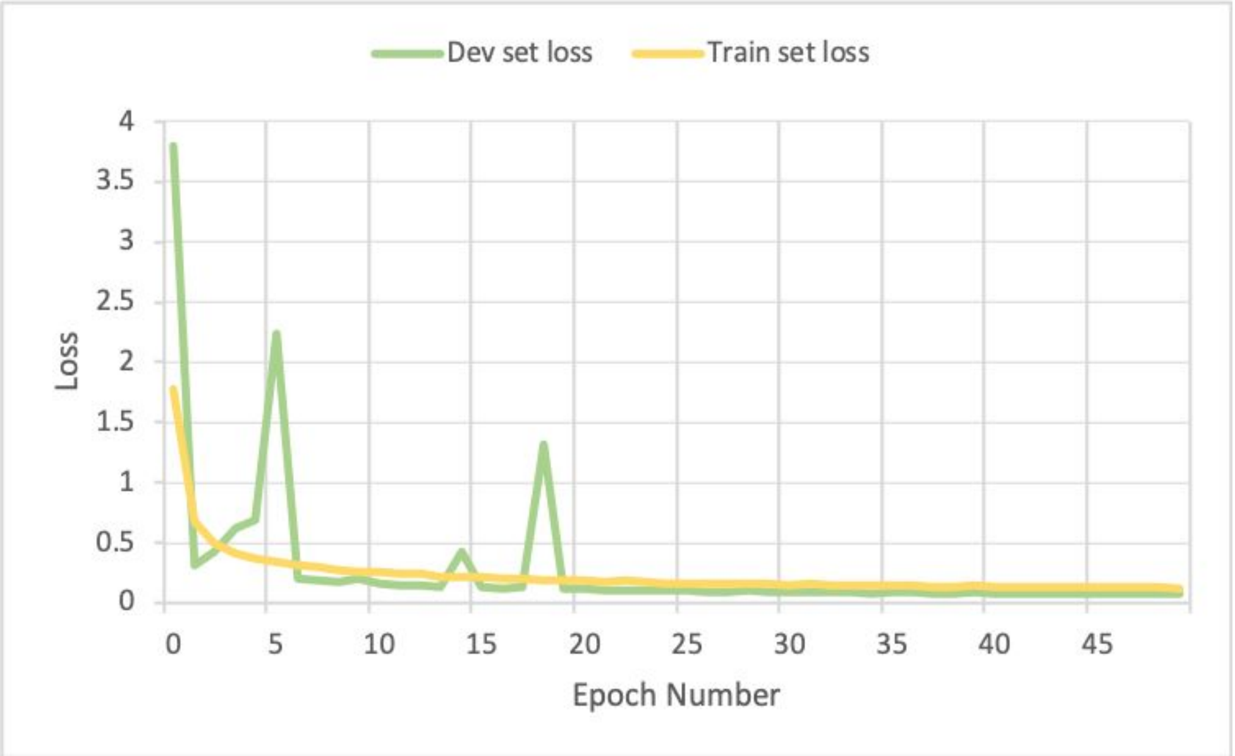


Normal Scale Confusion Matrix for Majority Vote

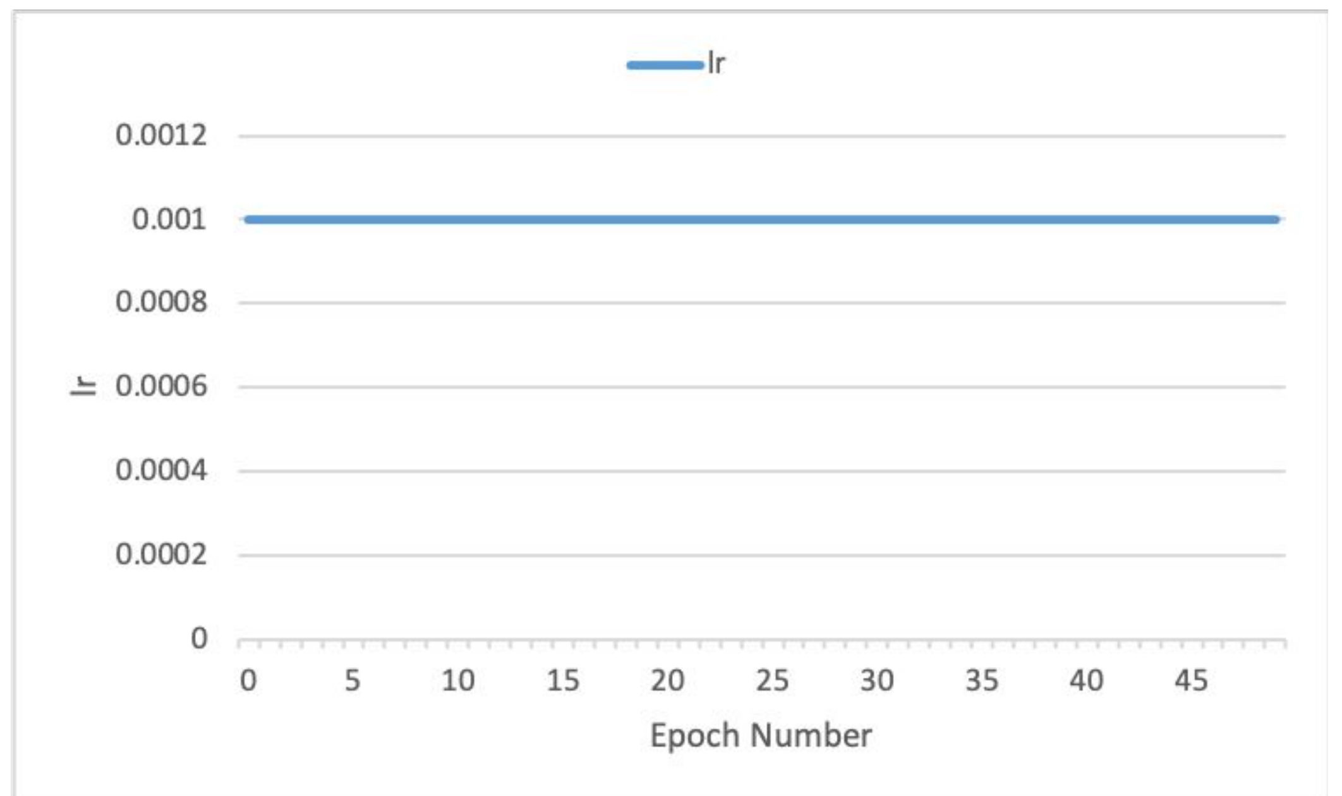
label \ prediction	backwards	boat	carry	down	end_comm	five	four	here	mosaic	num_delimiter	one	photo	start_comm	three	trueneg	two	up
backwards	50	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
boat	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
carry	0	0	65	0	0	0	0	0	0	0	0	0	0	0	0	0	0
down	0	0	0	42	0	0	0	0	0	0	0	0	0	0	0	0	0
end_comm	0	0	0	0	1.2e+02	0	1	0	0	0	0	0	2	0	0	0	0
five	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0
four	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0
here	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0
mosaic	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0
num_delimiter	0	0	0	0	0	0	0	0	0	89	0	0	0	0	1	0	0
one	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0
photo	0	0	0	0	0	0	0	0	0	0	0	82	0	0	2	0	0
start_comm	0	0	0	0	0	0	0	0	0	0	0	0	1.7e+02	0	0	0	0
three	0	0	0	0	0	0	0	0	0	0	0	0	0	35	0	0	0
trueneg	1	5	1	0	2	0	0	2	0	1	0	0	7	1	6.4e+02	0	1
two	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	36	0
up	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32

Linear scale confusion matrix

Resnet 18 with image size 320 x 240 on dataset2500 (our best single model) training process:



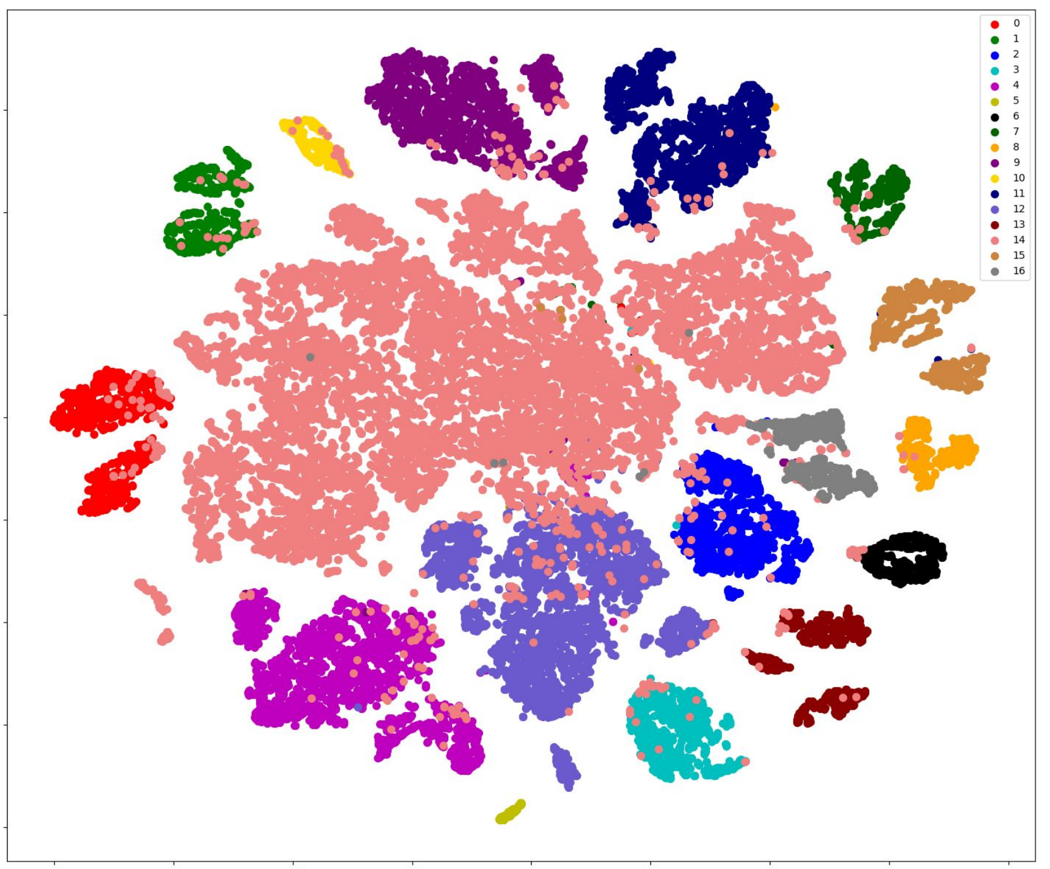
Loss for train set and dev set over the epochs
The image indicates the model doesn't overfit



Learning rate decided by ReducingLROnPlateau method in keras.callbacks

Discussion

- In spite of the good performance of the overall model, precision and recall of some classes can be unsatisfactory. It is mainly because the imbalance of the test set.



- tSNE analysis
The plot shows different classes are well divided by embedding. A number of *truenegative* samples have been classified as other classes, causing the precision of those classes drops, especially for classes with small sample size.

- Through experiments, our model can provide some insights for other underwater image classification problems: 1) image with lower resolution may serve as better input since it can weaken the effect of noise; 2) combining over-sampling and adding class-weight in the loss function can make up for the training set imbalance when data collection is not an option.

Future

Responding to the error analysis, three directions can followed:

GAN Augmentation: Using GAN to generate more CADDYIAN images for training.

Nested CNN - binary to multi-classifier: A nested CNN with a binary classifier picking up class *trueneg* followed by a multi-class classifier for labeling the rest of the images to other classes.

Nested CNN - localization prior to classification: Implementing a hand-localization CNN prior to the classification CNN.

References

[1] Chiarella, D., Bibuli, M., Bruzzone, G., Caccia, M., Ranieri, A., Zereik, E., ... & Cutugno, P. (2018) A novel gesture-based language for underwater human–robot interaction. Journal of Marine Science and Engineering 6(3), 91.

Acknowledgements

Thanks to everyone on the CS230 teaching team who provided this amazing course. Special thanks to Jo Chuang. As our project TA, he offered us a lot of valuable inspiration and help.