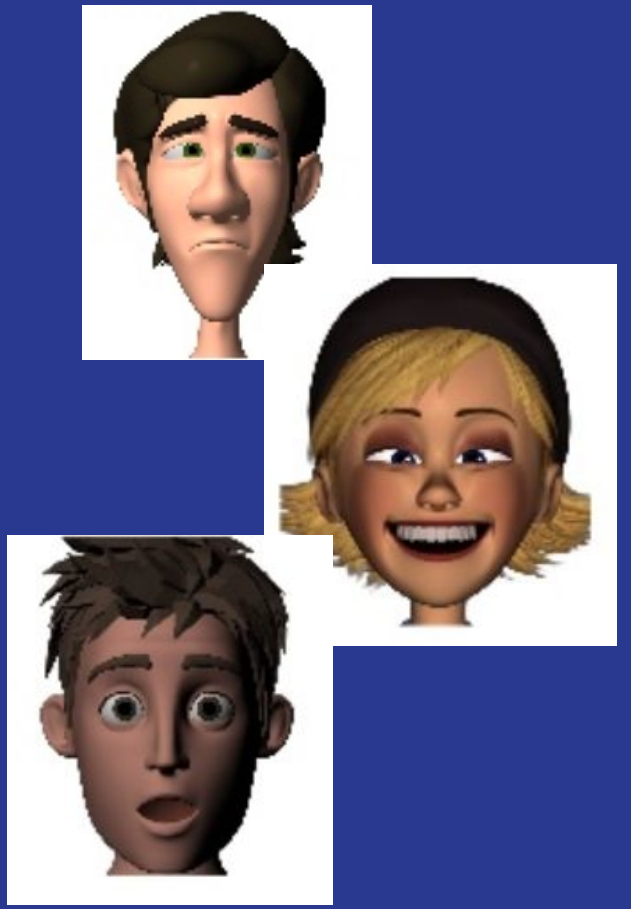
Emotions on Image Classification by Reinforcement Learning with Two-State Q-Learning

Ilona Tomkowicz

Goal: enhancement of traditional deep learning emotions classification

Approach	Accuracy
ResNet50	.8242
Proposed Approach using ResNet50	**.8309**
Inception V3	.8564
Proposed Approach using InceptionV3	**.8644**



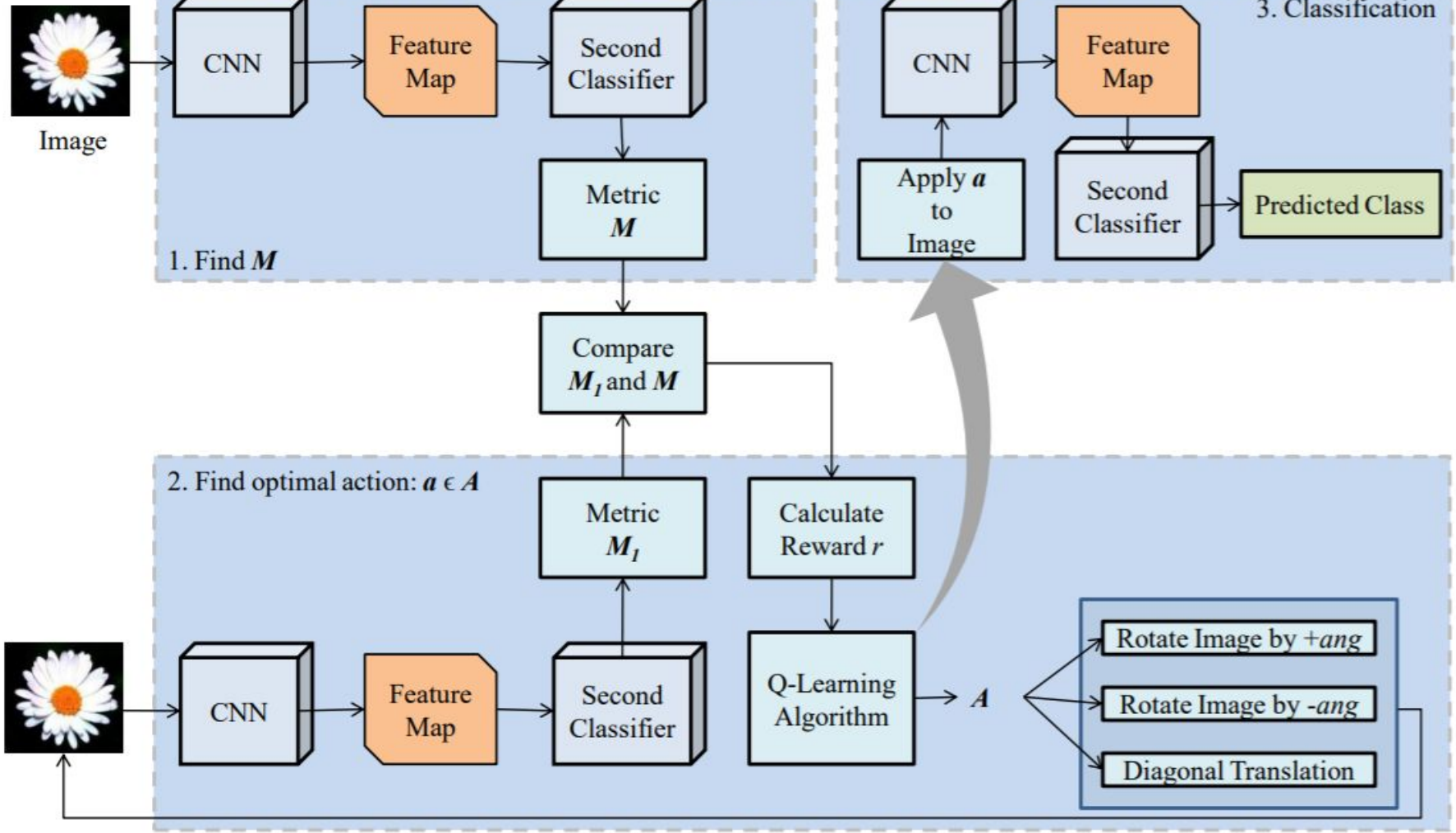
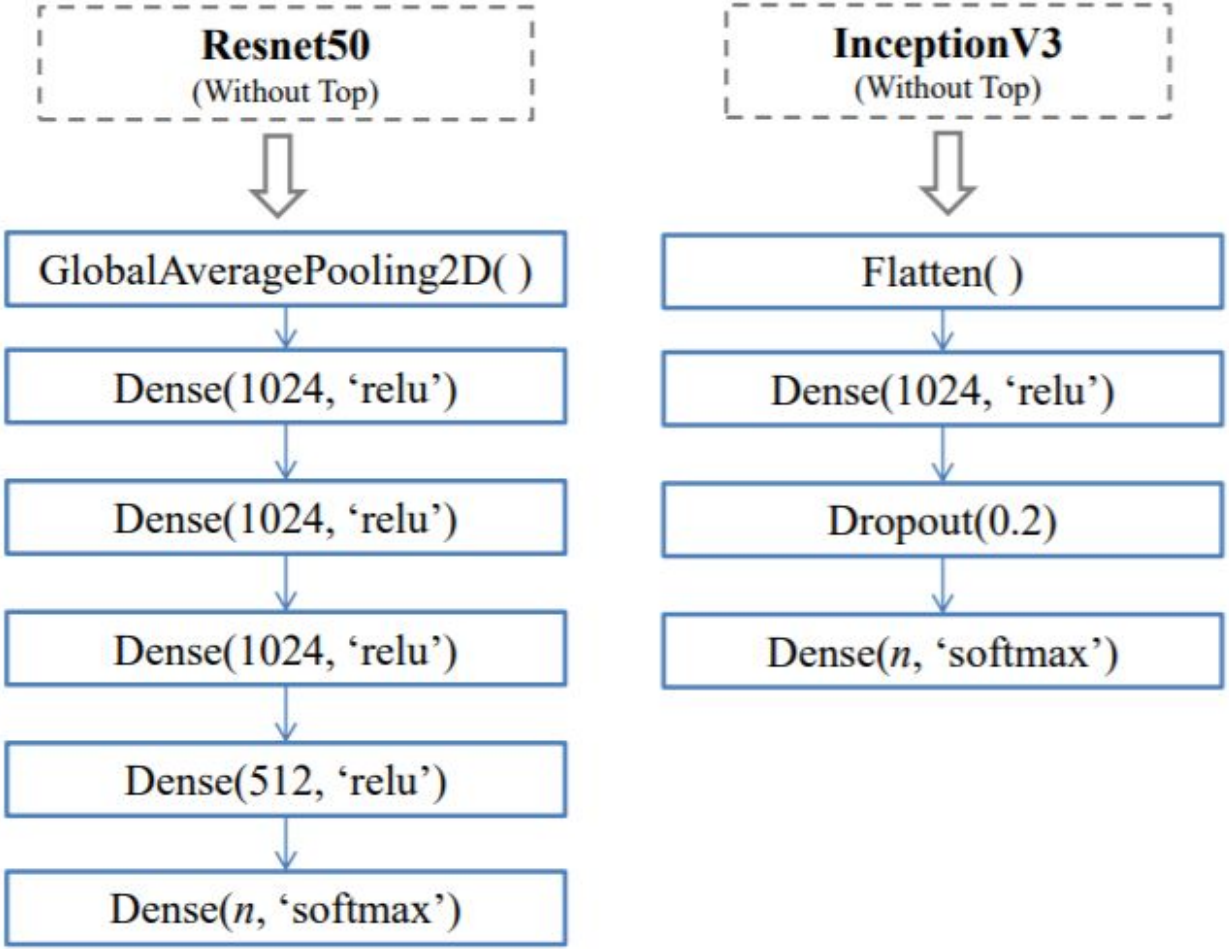


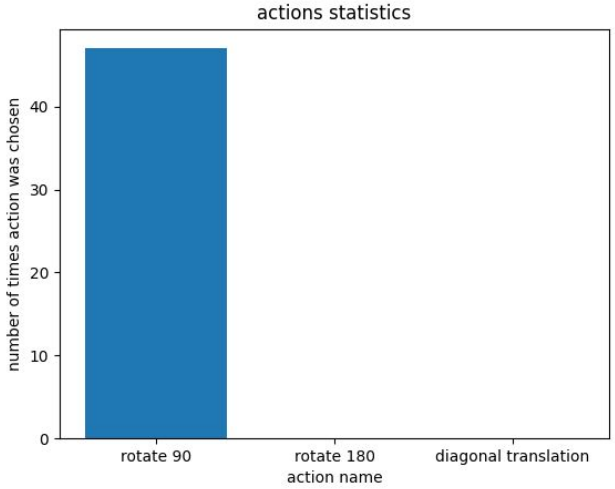
Image
CNN
Feature Map
Second Classifier
Metric M
1. Find M
3. Classification
CNN
Feature Map
Apply a to Image
Second Classifier
Predicted Class
Compare M1 and M
2. Find optimal action: a ϵ A
Metric M1
Calculate Reward r
Image
CNN
Feature Map
Second Classifier
Q-Learning Algorithm
A
Rotate Image by +ang
Rotate Image by -ang
Diagonal Translation

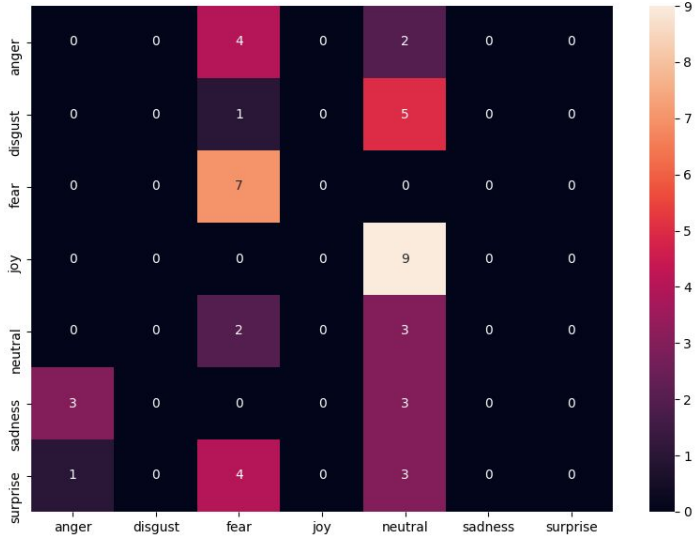

Resnet50
(Without Top)
GlobalAveragePooling2D( )
Dense(1024, 'relu')
Dense(1024, 'relu')
Dense(1024, 'relu')
Dense(512, 'relu')
Dense(n, 'softmax')
InceptionV3
(Without Top)
Flatten( )
Dense(1024, 'relu')
Dropout(0.2)
Dense(n, 'softmax')

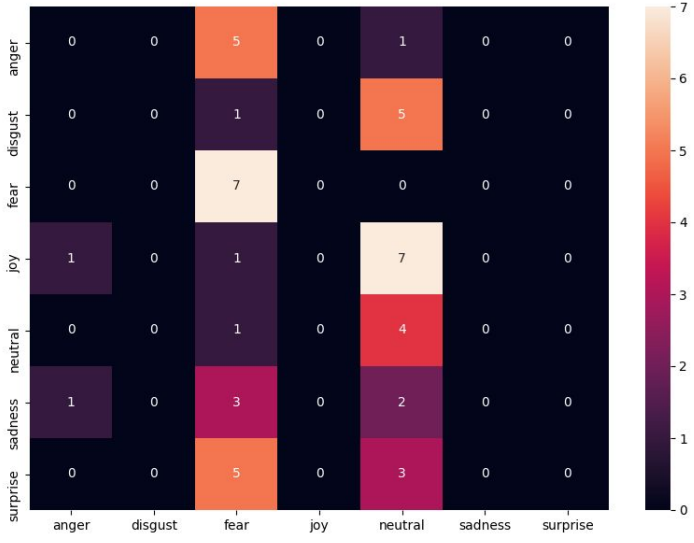
Experiment results

ResNet, 350 samples, Acc: 21,27 -> 23,40%

Actions: rotation 180 deg, rotation 90 deg, diag. translation

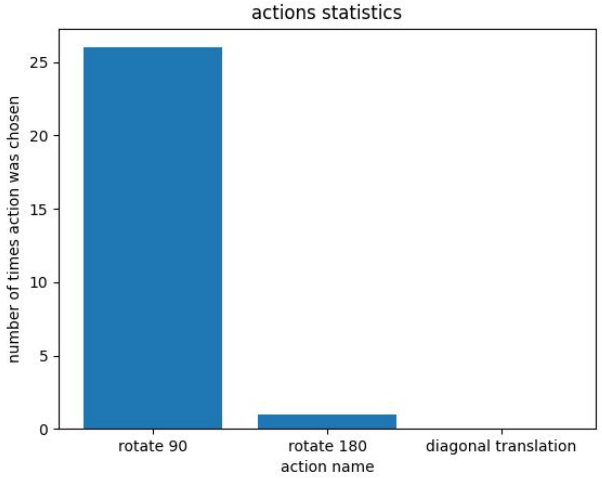


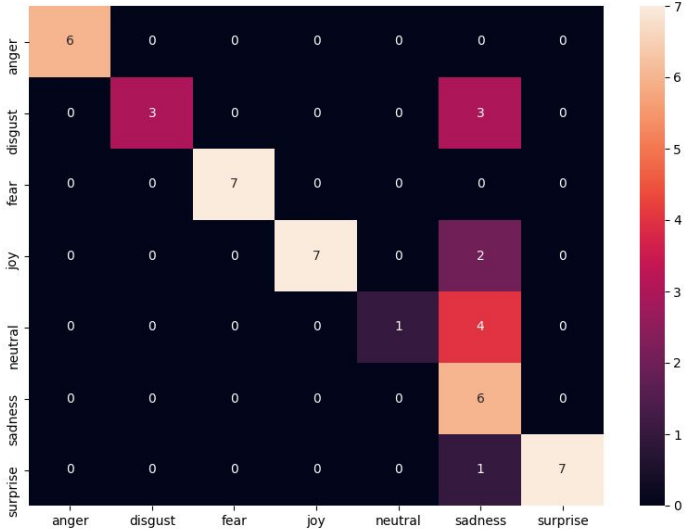


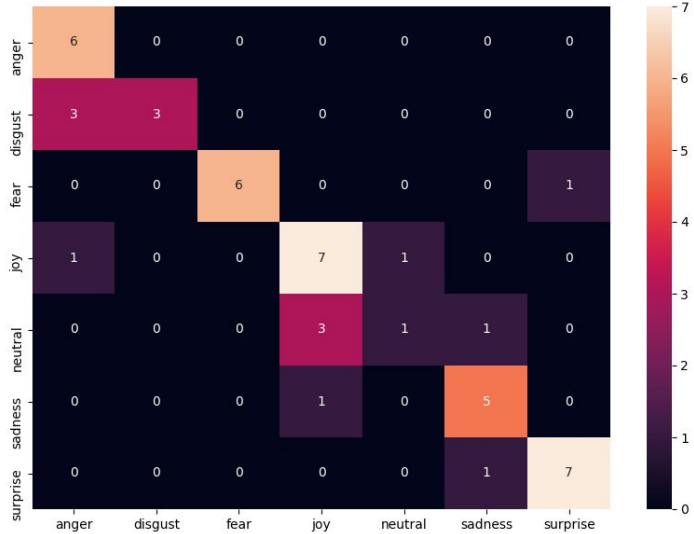


ResNet, 3500 samples, Acc: 93,64 -> 95,05%

Actions: rotation 180 deg, rotation 90 deg, diag. translation

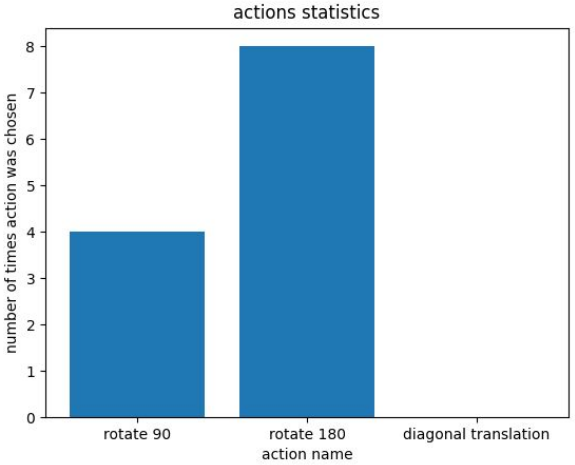


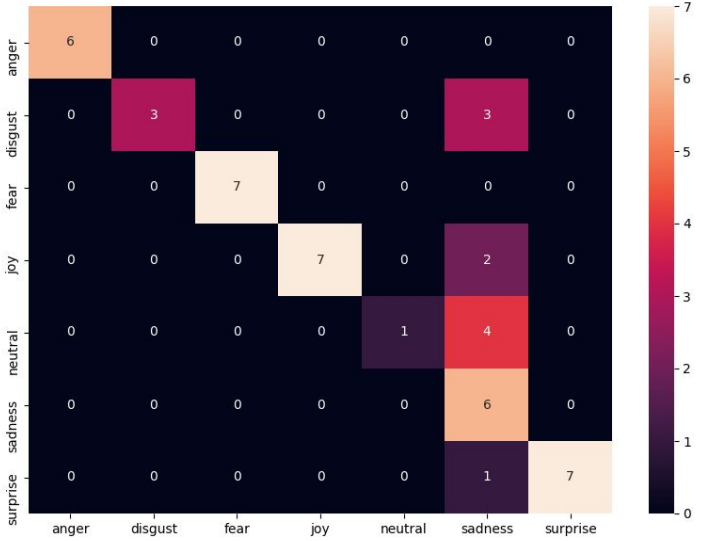


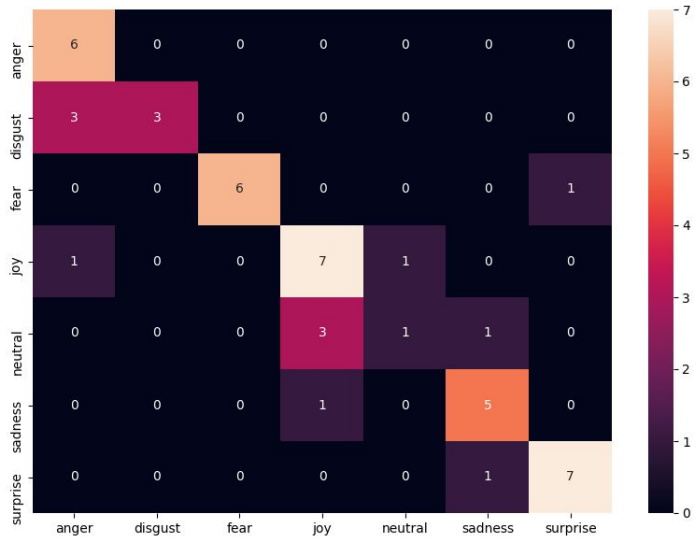


Inception, 350 samples, Acc: 74,46 -> 78,72%

Actions: rotation 180 deg, rotation 90 deg, diag. translation

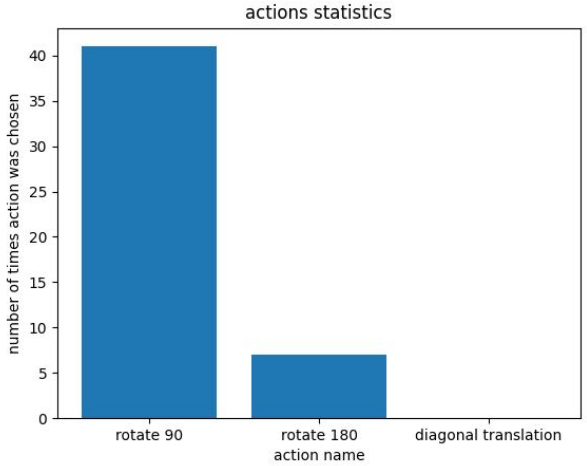


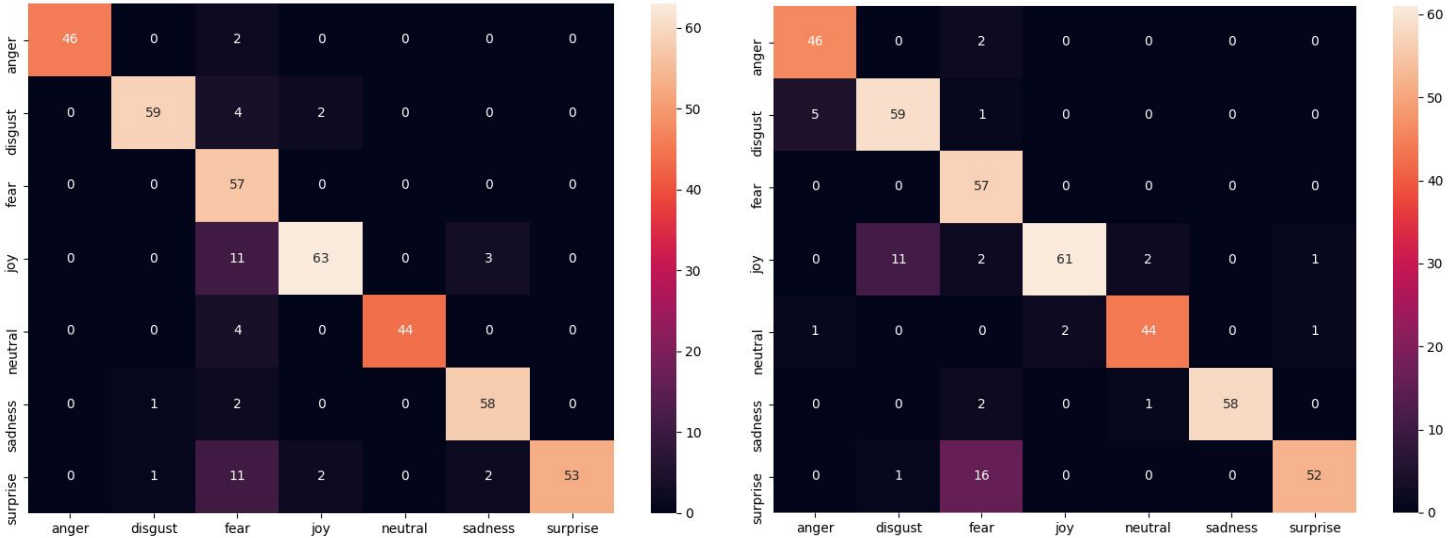




Inception, 3500 samples, Acc: 88,70 -> 89,41%

Actions: rotation 180 deg, rotation 90 deg, diag. translation





Too good to be true

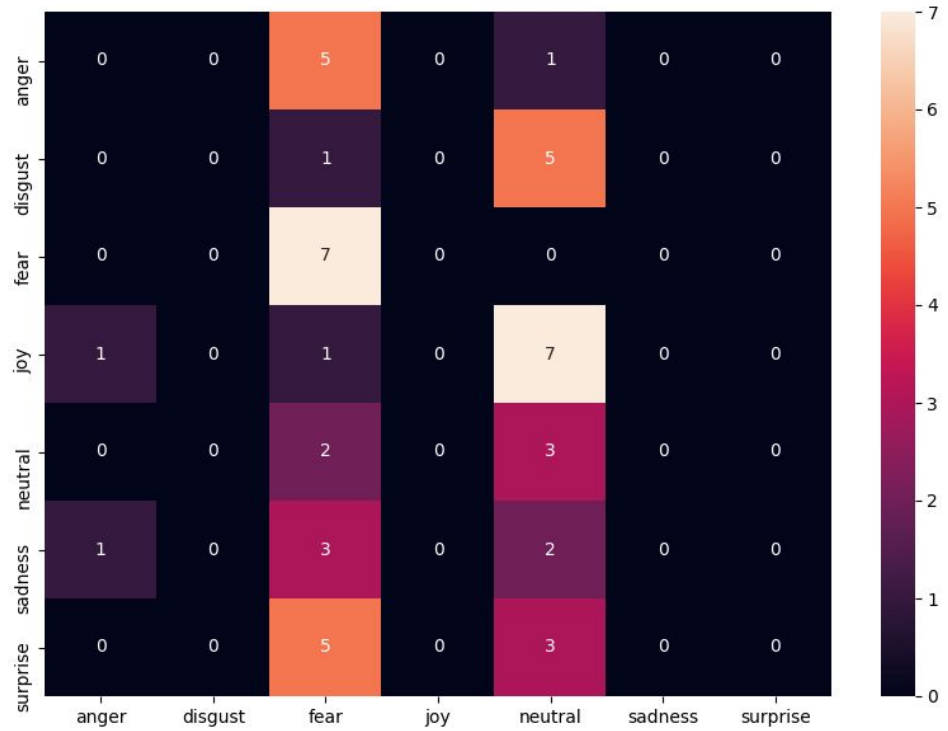
Algorithm evaluation methodology

- Evaluation used in article introduces conditional probability
- Assumes we know the test grand truth labels during the algorithm execution

1) Use traditional CNN (e.g. ResNet) to define a prediction on a test sample
2) If and only if CNN prediction is not equal to GT label - use RL enhancement
3) If RL was used, treat it’s result as the final response, else CNN response is final

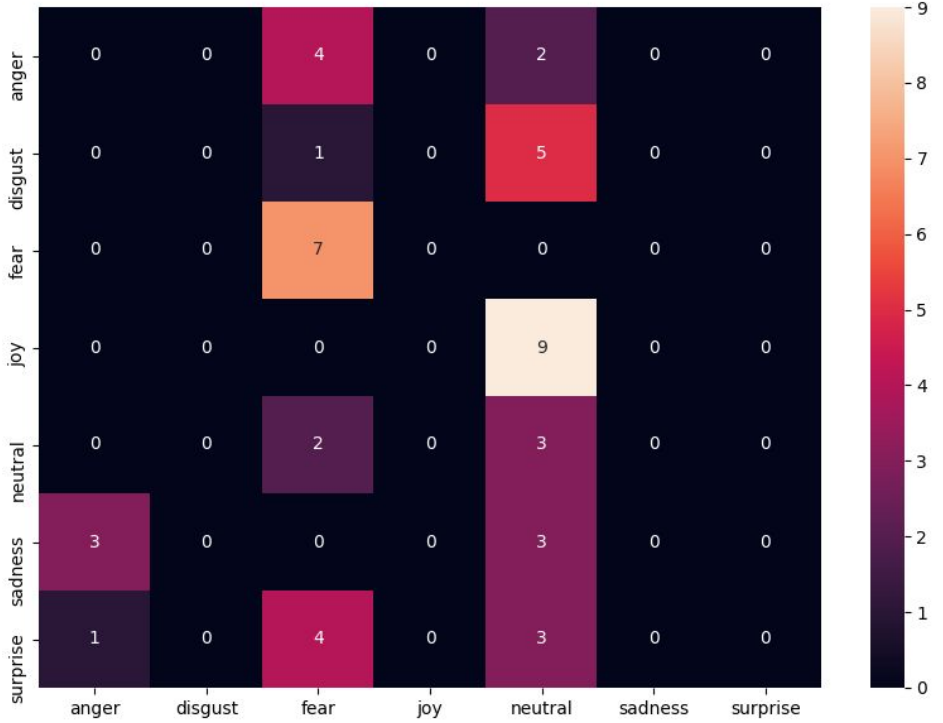
Adjustment:

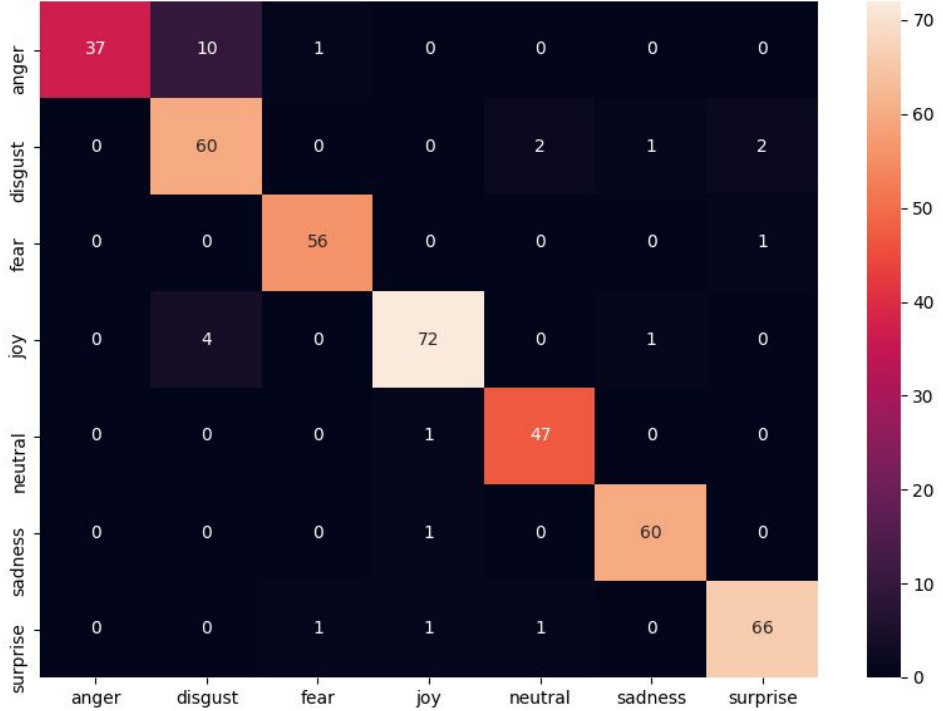
RL for all test samples



ResNet, 350 samples, Acc: 21 -> 21%

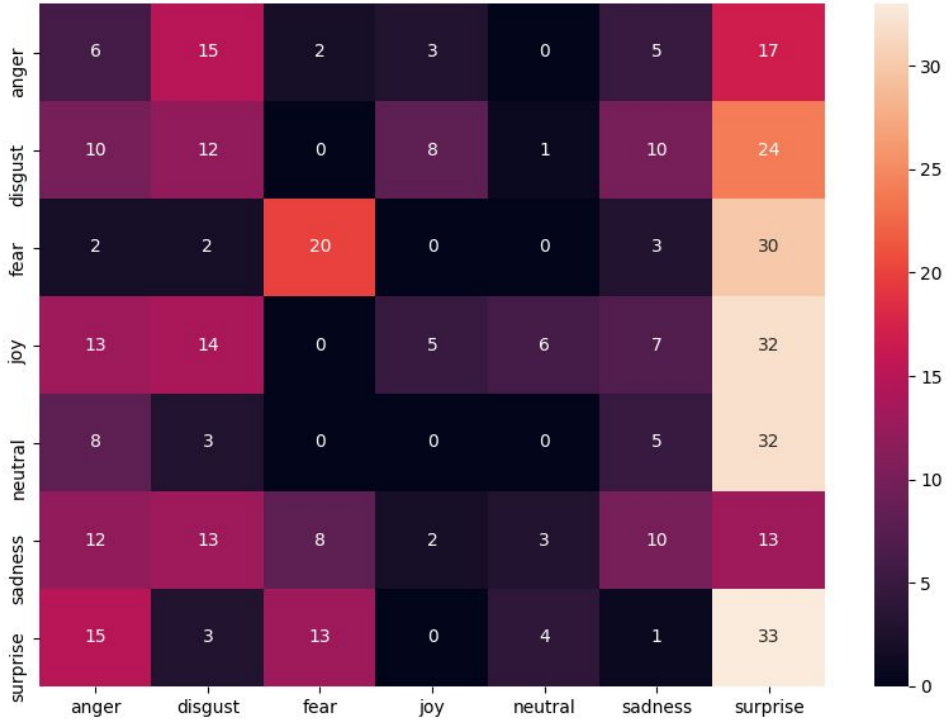
rotation 180 deg, rotation 90 deg, diag. translation

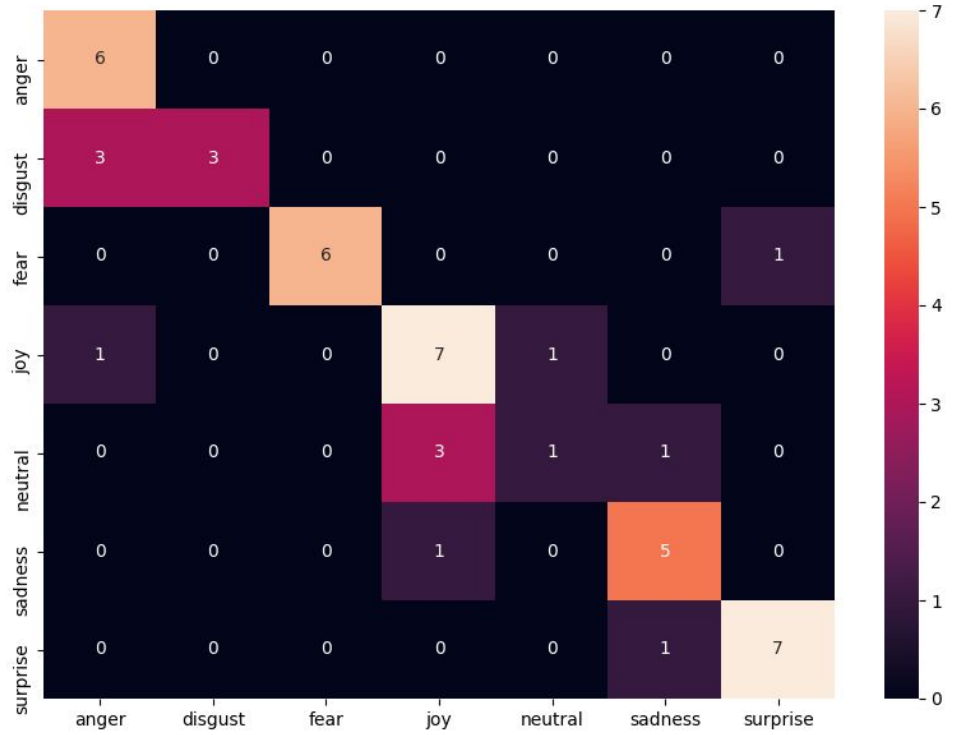




ResNet, 3500 samples, Acc: 94 -> 20%

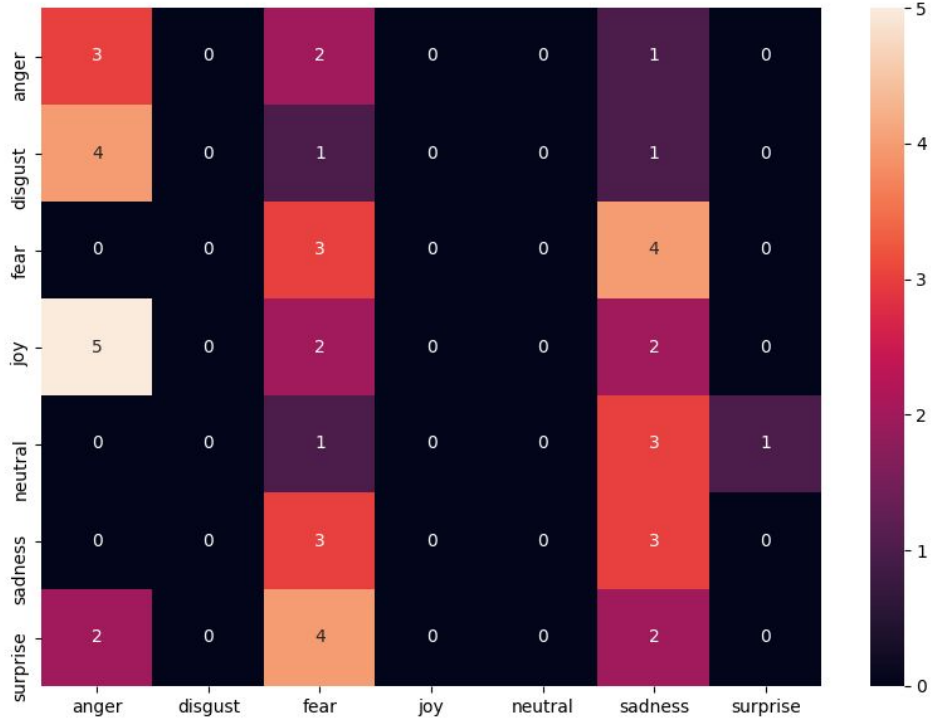
rotation 180 deg, rotation 90 deg, diag. translation

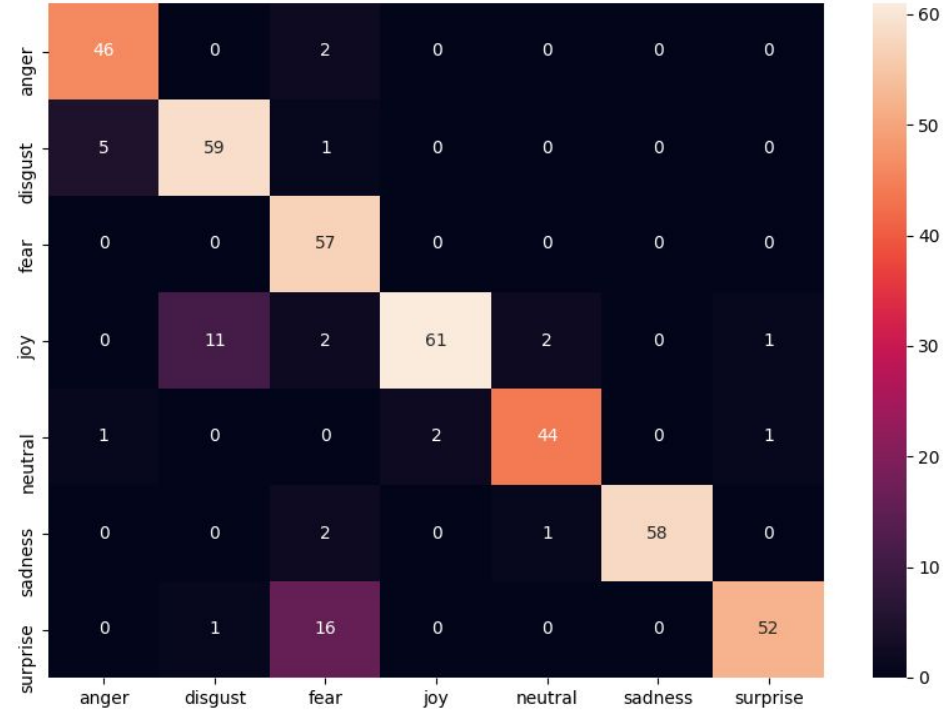




Inception, 350 samples, Acc: 74 -> 19%

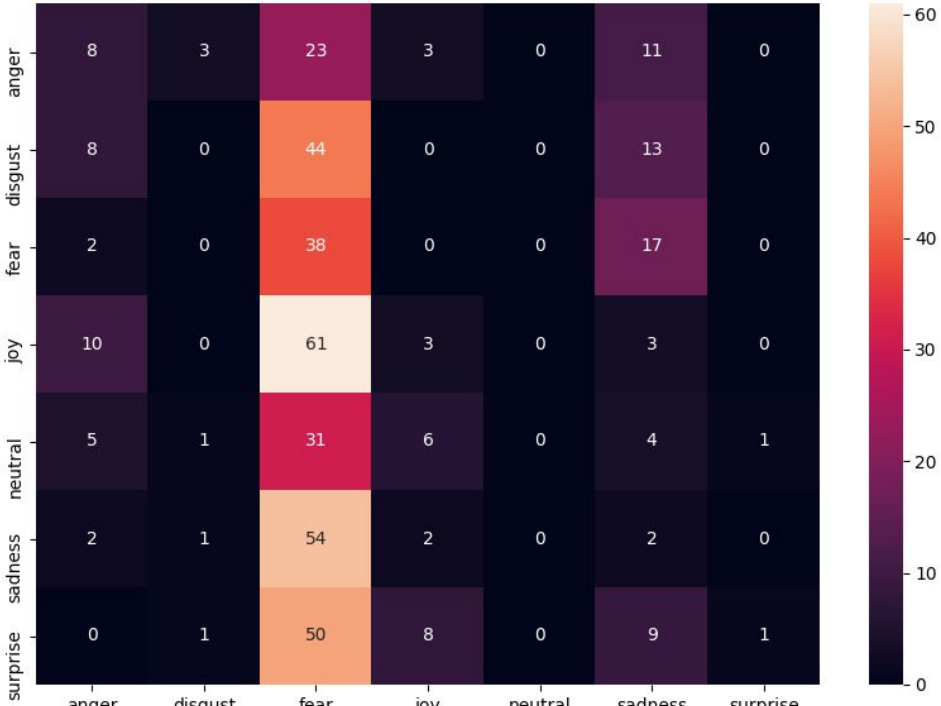
rotation 180 deg, rotation 90 deg, diag. translation





Inception, 3500 samples, Acc: 89 -> 12%

rotation 180 deg, rotation 90 deg, diag. translation



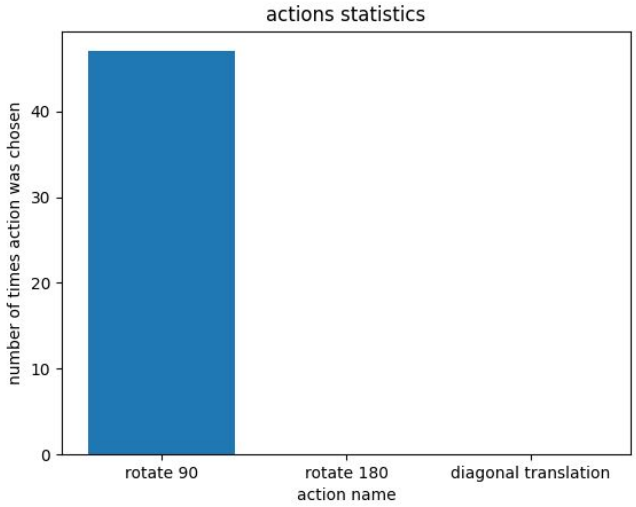
Original actions from article

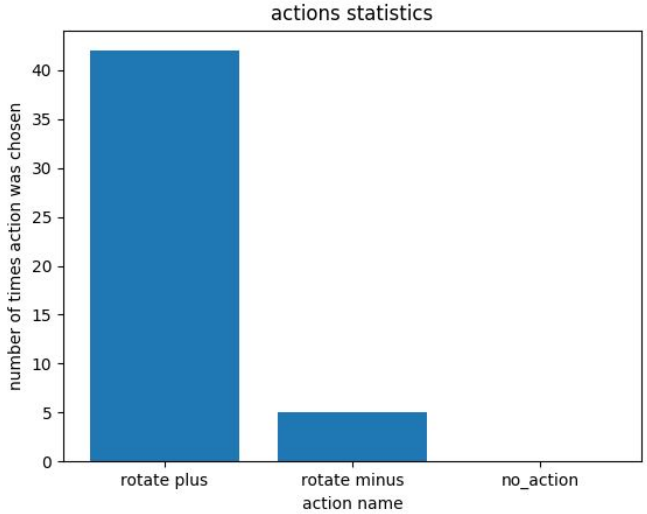
- Rotation 90 degrees
- Rotation 180 degrees
- Diagonal translation right-down 15 pixels

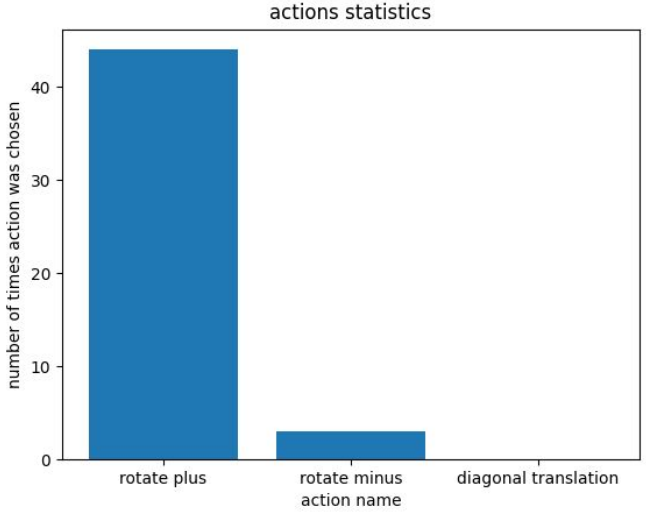
Do they make sense?







actions statistics
number of times action was chosen
0
10
20
30
40
rotate 90
rotate 180
diagonal translation
action name


actions statistics
number of times action was chosen
0
5
10
15
20
25
30
35
40
rotate plus
rotate minus
no_action
action name


actions statistics
number of times action was chosen
0
10
20
30
40
rotate plus
rotate minus
diagonal translation
action name

Actions comparison - demo

Actions	Rot90, rot180, diagTran	Rot+10, Rot-10, diagTran	Rot+10, Rot-10, noAction
Acc change [%]	0	2	2

- 90 and 180 angles do not make sense
- Diagonal translation never works better than no action
- Neither no action nor diagonal translation enhance classification

Adjustment:

4 actions: positive and negative angles for two small values

Actions	ResNet 350 samples, Acc 21%	ResNet 3500 samples, Acc 94%	Inception 350 samples, Acc 74%	Inception 3500 samples, Acc 89%
0: rotation 180 deg, 1: rotation 90 deg, 2: diag. translation	Acc: 21 % Optimal actions: 0: 100%	Acc: 20% Optimal actions: 0: 100%	Acc: 19% Optimal actions: 0: 90%, 1: 10%	Acc: 12% Optimal actions: 0: 95% 1: 5%
0: rotation +10 deg 1: rotation -10 deg, 2: diag. translation	Acc: 23% Optimal actions: 0: 90% 1: 10%	Acc: 20 % Optimal actions: 0: 100%	Acc: 47 % Optimal actions: 0: 95% 1: 5%	Acc: 49 % Optimal actions: 0: 99% 1: 1%
0: rotation +10 deg 1: rotation -10 deg, 2: no action	Acc: 23% Optimal actions: 0: 85% 1: 15%	Acc: 52% Optimal actions: 0: 98% 1: 2%	Acc: 67% Optimal actions: 0: 80% 1: 20%	Acc: 49 % Optimal actions: 0: 99% 1: 1%
0: rotation +5 deg 1: rotation -5 deg, 2: rotation +10 deg, 3: rotation -10 deg	Acc: 23 % Optimal actions: 0: 85% 1: 10% 3: 10%	Acc: 71% 0: 98% 1: 1% 3: 1%	Acc: 64% 0: 85% 1: 10% 3: 5%	Acc: 68% Optimal actions: 0: 95% 1: 5%
Action space search +5/-5/+10/-10/no change	Acc: 21%	Acc: 54%	Acc: 51%	Acc: 62%

States

States in the article are strictly connected with the reward in each step.

No adjustment was made here, even though it is not a good practice.

$$r = \begin{cases} +1, & if\ \boldsymbol{M_1} > \boldsymbol{M} \\ 0, & if\ \boldsymbol{M_1} = \boldsymbol{M} \\ -1, & if\ \boldsymbol{M_1} < \boldsymbol{M} \end{cases}$$

State 1 when action enhanced std (reward >= 0)

State 0 otherwise.

Conclusions:

- increasing image features std does not enhance CNN classification,
- the article contains a deceiving algorithm evaluation method.

Literature

“Image Classification by Reinforcement Learning with Two-State Q-Learning” Abdul Mueed Hafiz, 2020
FERG DB: https://grail.cs.washington.edu/projects/deepexpr/ferg-2d-db.html