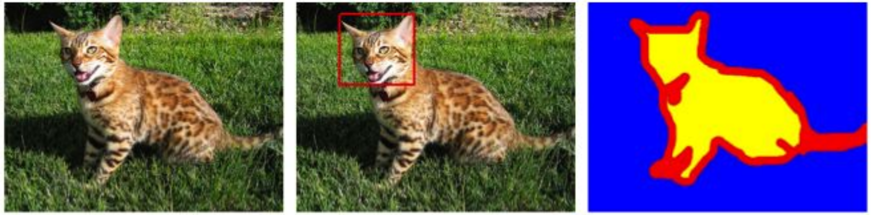
COGS 225 PROJECT

Dog/Cat Breed Image
Classification/Localization/Segmentation with Transfer Learning



[1]

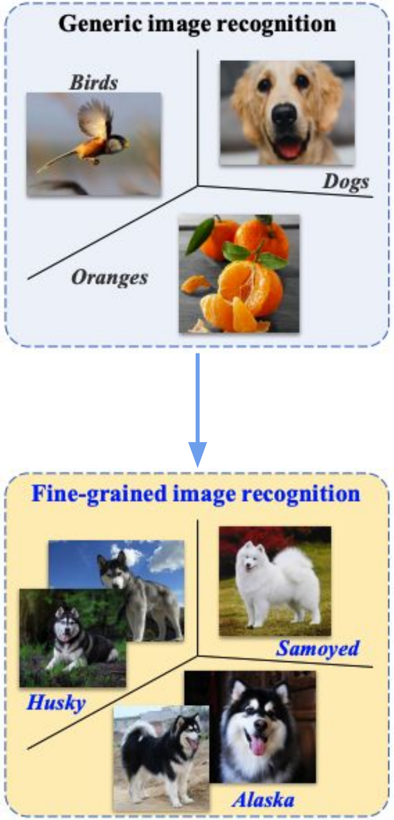
Haenara Shin - A53233226

BACKGROUND

(-) Classification of breeds in dog or cat is hard to non-expert.
(-) Some breeds are not familiar with people.
(-) Fine-grained image recognition is more challenging.

Motivation: There is a need for automated classification method.
Solution & Goal: Dog/Cat image classifier model.
(+) Extra-trial experiment: Based on classification, building the localization (multi-task learning) and segmentation for images.

(+) Potential Real-world Application
: Helping non-experts to distinguish different species/breeds.
: Object detection, Products recognition, mammal classification...



[2]

PROBLEM STATEMENT

Problem
: In order to address the disadvantages of classification for species/breed images, the efficient automatic classification model should be implemented.

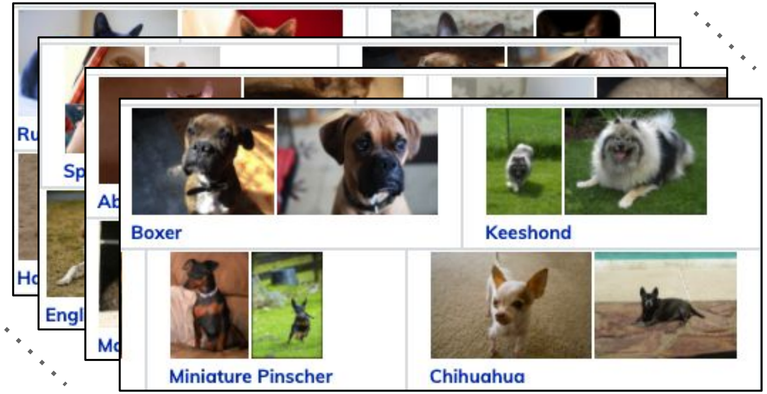
Solution: Building classification models from Convolutional neural networks
Goal

1. **Object Classification:** Given dataset, predict 37 breed classes.
2. **Object Localization:** The head images can be localized.
3. **Object Segmentation:** The image can be segmented among foreground and background.

DATASET

- 7349 dog and cat images of 37 breeds (Oxford-IIIT Pet Dataset): http://www.robots.ox.ac.uk/~vgg/data/pets/
- Roughly 200 images per each breed with
  (i) associated ground truth annotations of species and breed name,
  (ii) a tight bounding box ROI around the head, and
  (iii) a pixel level foreground-background segmentation.

- Dog breeds/images: 25/4978
- Cat breeds/images: 12/2371
- The images have large variations in scale, pose and lighting.
- 75% Training set vs. 25% Validation set



[1]

(1) Classification

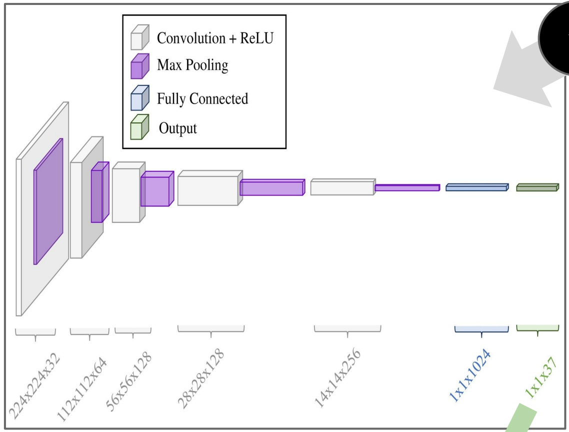
1) Preliminary Test

a) Vanilla CNN
b) Learning rate schedule (Exponential decay)
c) Data augmentation (Cut mix)

2) Transfer learning

Classification - Prelim. Test

Vanilla CNN Structure



(5920,224,224,3)

(1458,224,224,3)

7378 images

1
- Poor performance (Acc.): 26.75%

*20 epoch

2
- + BatchNormailzation
  + Learning rate schedule
  (Exp. decay) [3]

2
- Improved, but still low performance (Acc.: 31.34%)

3
- + Data augmentation (Cut Mix)[4]

3
- Worse performance (Acc.) : 29.01%

*50 epoch

4
- Transfer learning (MobileNetV2) + Exp. decay

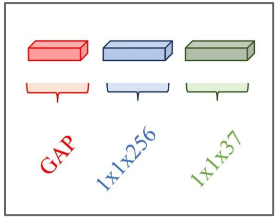
- Good performance (Acc.): 93.76%
- Significantly Improved!
  ⇒ Models should be changed.

*20 epoch

Classification - Transfer Learning (Model)

- 10 pre-trained models are prepared from TensorFlow framework

VGG-16
VGG-19
InceptionV3 (IncpV3)
Xception (Xcep)
ResNet50 (RN50)
ResNet152V2 (R152V2)
MobileNetV2 (MbNv2)
NASNetMobile (NASMo)
DenseNet121 (DN121)
InceptionResNetV2 (InRNv2)



- Replaced Bottleneck Layer

: **Global Average Pooling** (GAP), **One Dense Layer** (BatchNormalization, ReLU), and **One Dense Layer** (softmax)

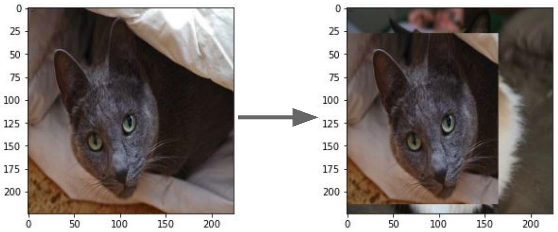
Classification - Transfer Learning (Results)

7378 images

(5920,224,224,3)

(1458,224,224,3)

Data augmentation (Cut mix)



Pre-trained models with Exp. decay learning

	Models									
	VGG16	**VGG19**	**MbNV2**	**IncpV3**	**Xcep**	**DN121**	**RN50**	**R152V2**	**NASMo**	**InRNV2**
Acc. [%]	89.85	91.22	93.28	93.62	**94.17**	**94.51**	92.11	91.02	92.87	93.48
***T [s]**	50	58	**29**	**30**	68	42	38	80	54	68

*50 epoch
*T = ~avg. time/step

Observation: **DenseNet121** shows the **best accuracy**, but the **best cost-efficient** model is **MobileNetV2**.

(2) Localization

1) Preliminary Test

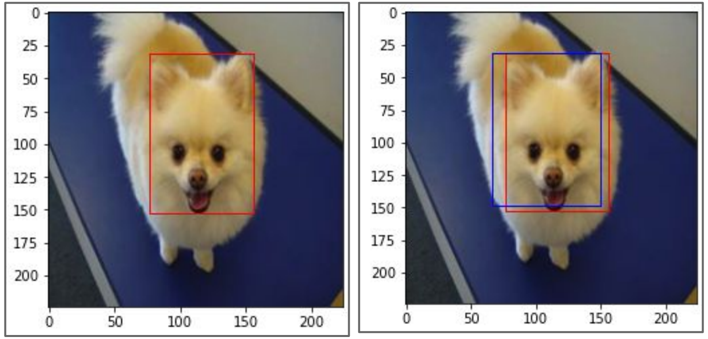
: Vanilla CNN, and Top 5 Accuracy Pre-trained models

: Class Activation Map [5] for 2-class vs. 37-class classification

2) Multi-task learning [6]

: Classification + Localization

: Top 5 Accuracy Pre-trained models



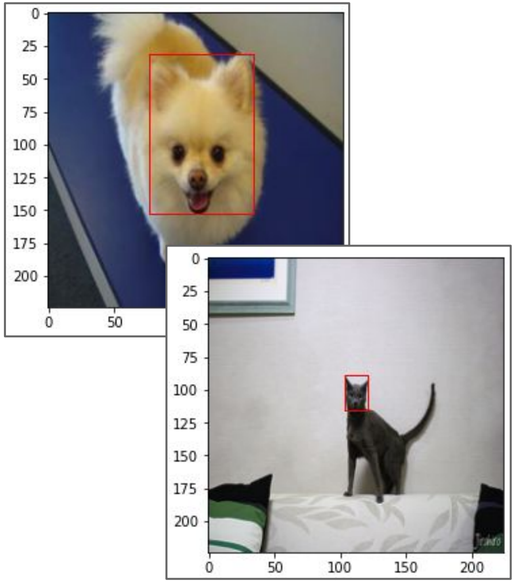
Localization - Prelim. Test (Models)

3685 annotated images

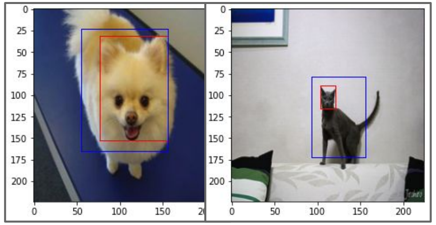
(3000,224,224,3)

(685,224,224,3)

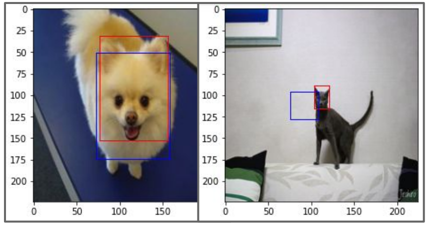
*40 epoch



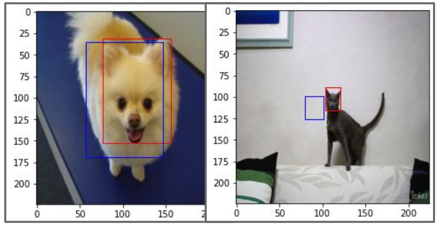
Vanilla CNN



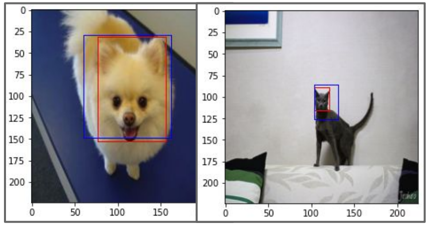
MobileNetV2



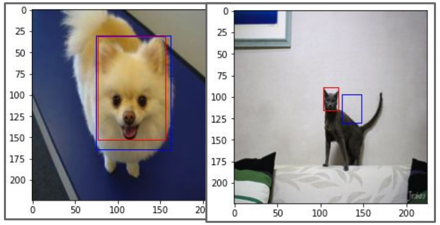
Xception



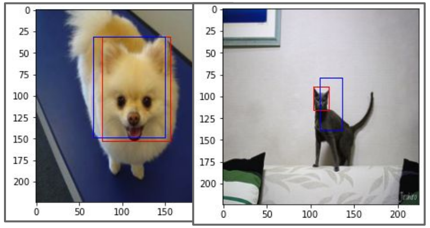
InceptionV3



InceptionResNetV2



DenseNet121



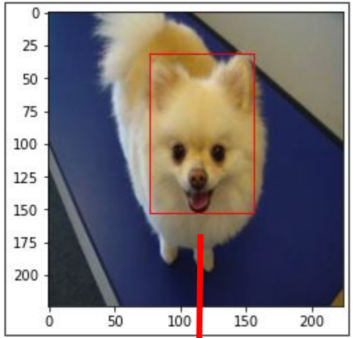
Model	IoU[%]
Vanilla	49.65
MbNv2	65.24
Xcep	62.05
Inv3	69.32
InRNv2	66.85
DN121	70.69

Multi-task Learning!

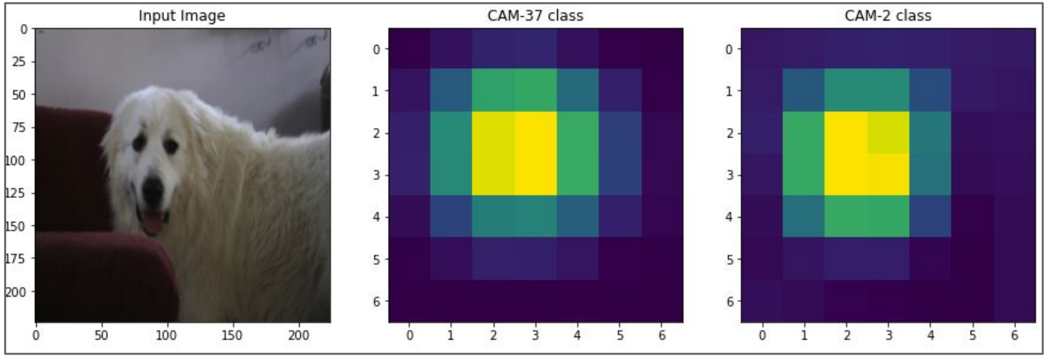
Localization - Prelim. Test (CAM)

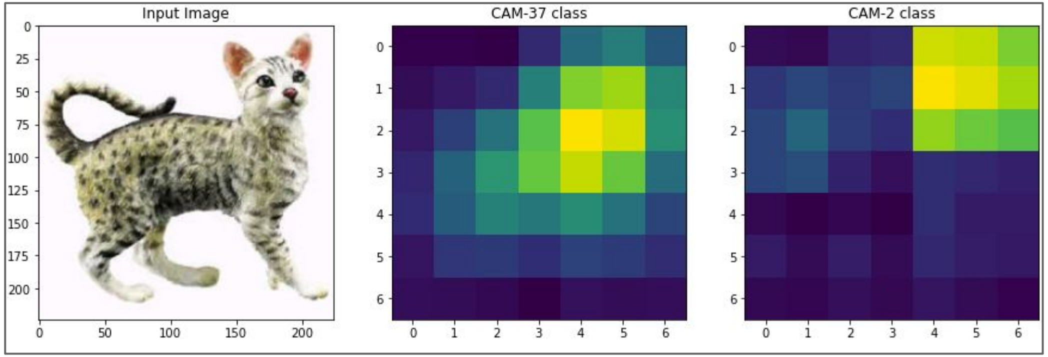
DenseNet121	37 classes (breeds)	2 classes (species)

More actively focusing on the 'local' region.

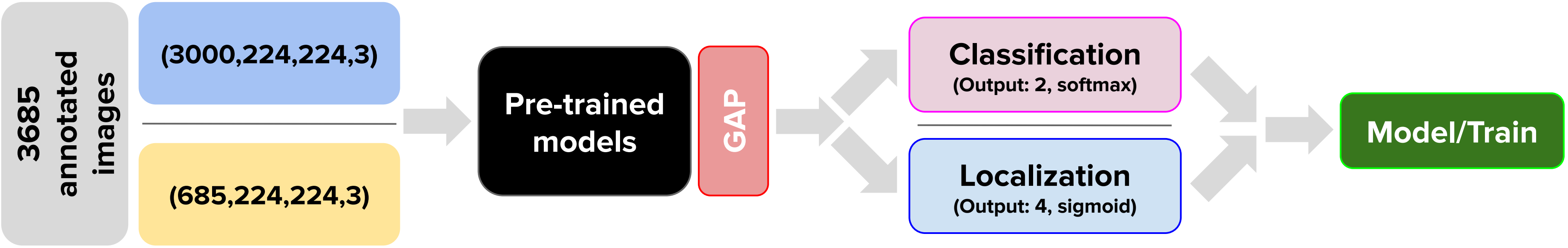


Head ROI is the most important for localization.

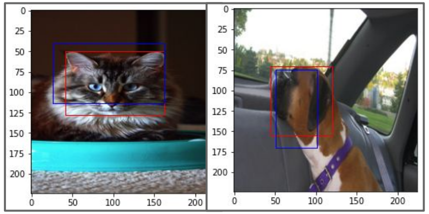




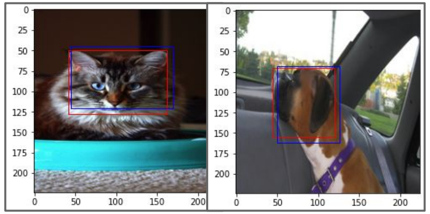
Localization - Multi-task Learning



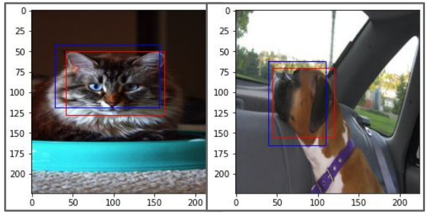
MobileNetV2



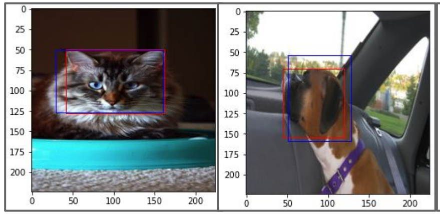
Xception



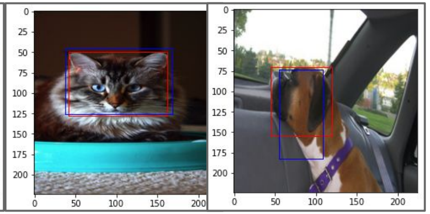
InceptionV3



InceptionResNetV2



DenseNet121



All IoU is enhanced!

Model	IoU[%]	*T [s]
MbNv2	71.94	15
Xcep	73.90	35
Incpv3	70.58	17
InRNv2	74.84	34
DN121	74.78	22

*40 epoch
*T = ~avg. time/step

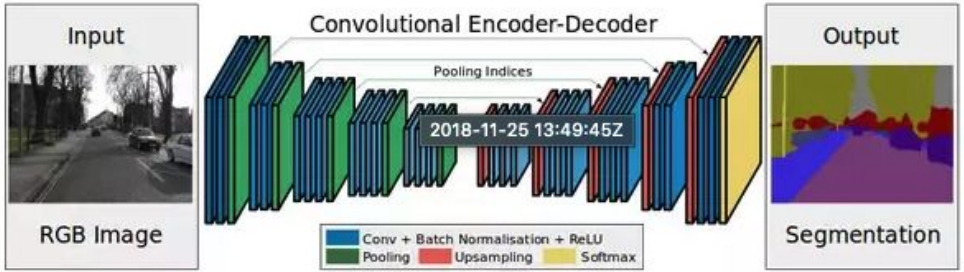
(3) Segmentation

1) Preliminary Test

: U-Net like model training through random initialization

2) Transfer Learning

: U-Net like model + Pre-trained VGG-16, VGG-19



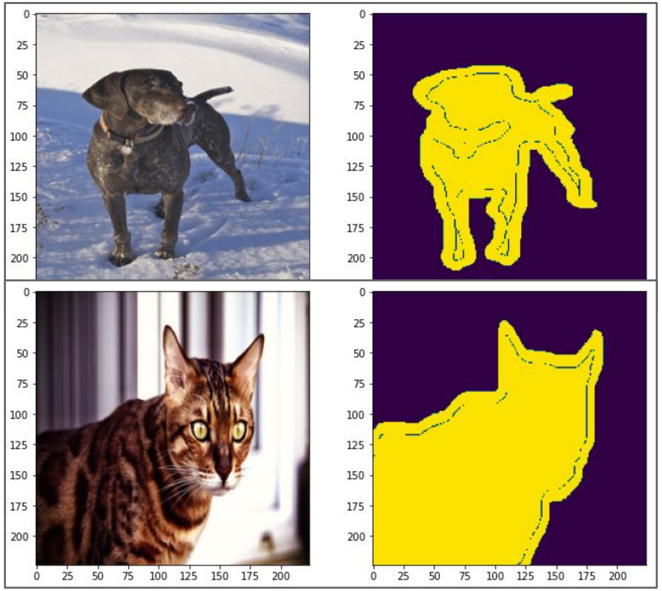
[7]

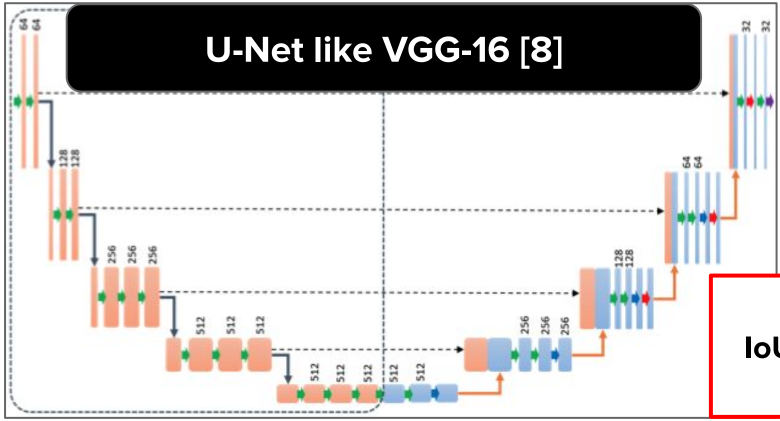
Segmentation - Prelim. Test

7378 images

(5920,224,224,3)

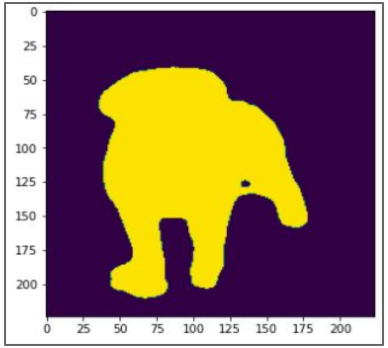
(1458,224,224,3)

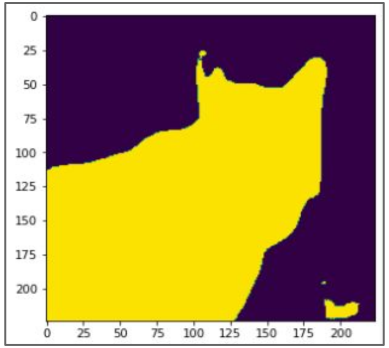




IoU: 83.39%

*20 epoch





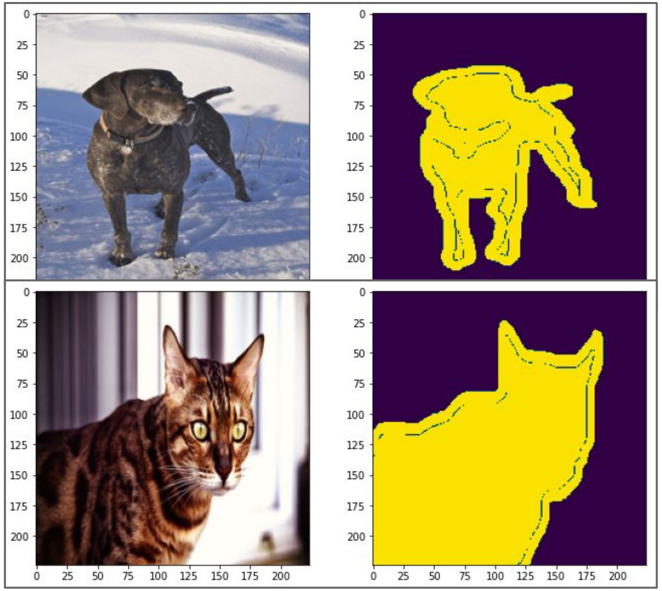
Transfer Learning!

Segmentation - Transfer Learning

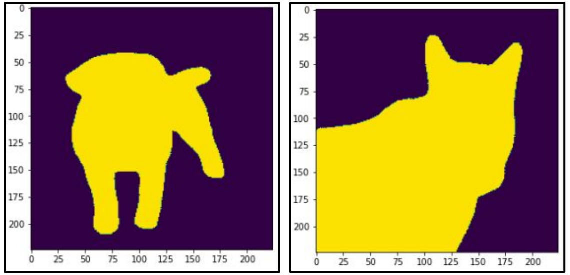
7378 images

(5920,224,224,3)

(1458,224,224,3)



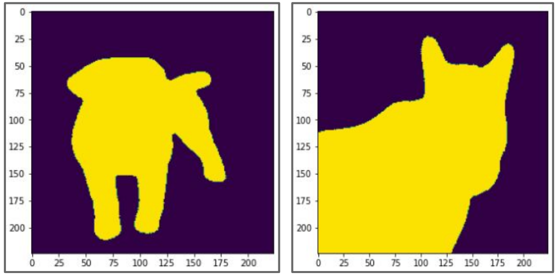
U-Net (like VGG-16) + VGG-16



IoU: 90.02%

*20 epoch

U-Net (like VGG-19) + VGG-19



IoU: 90.09%

*20 epoch

★ U-Net with pre-trained model's weight shows the improved and better performance

CONCLUSION & Future work

[Goal]

1. **Classification**: Classify 37 breeds.
2. **Localization**: Localize images' head region
3. **Segmentation**: Segment images as foreground and background.

[Solution]: CNN with Transfer Learning

1. **Classification**: Transfer learning + Exp. decay learning rate + Cut mix → Future work : Adaptive learning rate [3]
2. **Localization**: Multi-task learning (2-class Classification + Localization)
3. **Segmentation**: U-Net with Transfer learning

[Results]

1. **Classification**: DenseNet121 (Acc.), MobileNetV2 (Cost-efficiency)
2. **Localization**: DenseNet121 and InceptionResNetV2 (IoU), MobileNetV2 (Cost-efficiency)
3. **Segmentation**: U-Net with VGG-19 (IoU)

➔ Future work: Test more deeper or sophisticated pre-trained models.

★ **Every task with pre-trained models in image classification/localization/segmentation shows significantly improved or better performance (Acc. or IoU).**

REFERENCES

[1] http://www.robots.ox.ac.uk/~vgg/data/pets/

[2] Wei X S, Wu J, Cui Q. *Deep learning for fine-grained image analysis: A survey.* arXiv preprint arXiv:1907.03069, 2019

[3] https://towardsdatascience.com/learning-rate-schedules-and-adaptive-learning-rate-methods-for-deep-learning-2c8f433990d1

[4] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo, *CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features*, arXiv:1905.04899v2, 2019

[5] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba, *Learning Deep Features for Discriminative Localization,* CVPR, p. 2921-2929, 2016

[6] Sebastian Ruder, *An Overview of Multi-Task Learning in Deep Neural Networks*, arXiv:1706.05098v1, 2017

[7] https://towardsdatascience.com/review-segnet-semantic-segmentation-e66f2e30fb96

[8] Maayan Frid, Avi Ben, Rula Amer, Hayit Greenspa, *Improving the Segmentation of Anatomical Structures in Chest Radiographs using U-Net with an ImageNet Pre-trained Encoder*, arXiv:1810.02113, 2018