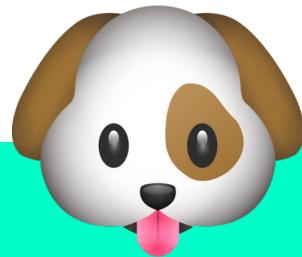


DOG BREED IDENTIFICATION



Zhonghe Han, Tianmin Li, Yezhu Li, Yiru Wang

WHAT KIND OF DOG IS THAT?

Determine the breed of a dog in an image



AGENDA

Data Description

Data Preparation

Computing Resources

Model Development

Model Deployment

Conclusion

DATA DESCRIPTION

Datasets include **a set of images of dogs**. Each image has a filename that is its unique id. The dataset comprises **120 breeds of dogs**. The goal of this project is to create a classifier capable of determining a dog's breed from a photo.

Some example in the list of breeds is as follows:

- Affenpinscher
- Afghan_hound
- African_hunting_dog
- Airedale
- American_staffordshire_terrier
- Appenzeller
- Australian_terrier
- Basenji
- Basset
- Beagle
- Bedlington_terrier

.....

DATA DESCRIPTION

Datasets include **a set of images of dogs**. Each image has a filename that is its unique id. The dataset comprises **120 breeds of dogs**. The goal of this project is to create a classifier capable of determining a dog's breed from a photo.

Some example in the list of breeds is as follows:

- Affenpinscher
- Afghan_hound
- African_hunting_dog
- Airedale
- American_staffordshire_terrier
- Appenzeller
- Australian_terrier
- Basenji
- Basset
- **Beagle**
- Bedlington_terrier

.....



DATA PREPARATION

Data size:

10,000 images in total, split data into 80% train / 20% validation

- Validation size: 2,045
- Train size: 8,177

Data preprocessing:

- **Scaling:** Divide the image matrix by 255, squeezing the pixel values to 0-1 range
- **Zero-Centered / Normalization**

COMPUTING RESOURCES

- Used 1 PC GPU (GeForce 1080ti)
- Model Training Time:
20–40 seconds/epoch, depending on
the input size and layers added

MODEL DEVELOPMENT

Step 1

Using Transfer Learning on different pre-trained models, pick the best fit

Step 2

Using extracted features to train add-on layers to get a model that fit our data better

Step 3

Evaluate models and pick out the one with highest validation accuracy

TRANSFER LEARNING MODEL SELECTION

Pre-trained models:

- VGG16
- Xception
- InceptionV3
- ResNet50
- Inception_v3 + Resnet152
- InceptionResnetV2

.....

VGG16

DataPrep	Pooling	Dense	Activation	Optimizer	Batch Size	epoch	early stop	Val-Acc(epoch n)	Train-Acc(epoch n)	Val_loss	Train_loss
preprocess	avg	[256, 120]	[relu, softmax]	adam	100	100	10 acc	67.9(55)	73(60)	0.29 / 27	0.70 / 25

XCEPTION

Base	DataPrep	Pooling	Dense	Activation	Optimizer	Batch Size	epoch	early stop	Val-Acc(epoch n)	Train-Acc(epoch n)
Xception	Centered	None	2: [256, 120]	[relu, softmax]	adam	20	30		67.8(96)	49.7(94)
Xception	rescale	avg	2: [256, 120]	[relu, softmax]	adam	100	100	10 acc	89.3(32)	84.6(42)
Xception	rescale	avg	2: [1024, 120]	[relu, softmax]	adam	100	100	10 acc	89.2(30)	84.3(40)
Xception	preprocess	avg	2: [256, 120]	[relu, softmax]	adam	100	100	10 acc	89.8(15)	81.2(24)
Xception	preprocess	avg	2: [1024, 120]	[relu, softmax]	adam	100	100	10 acc	89.5(9)	82.1(18)

INCEPTION_V3

DataPrep	Pooli ng	Dropo ut	Dense	Activation	Optimi zer	Batch Size	epoch	early stop	Val-Acc(epoch n)	Train-Acc (epoch n)
preprocess	avg	-	2: [256, 120]	[relu, softmax]	adam	100	100	10 acc	90.5(33)	75.6(39)
preprocess	avg	-	2: [1024, 120]	[relu, softmax]	adam	100	100	10 acc	88.6(21)	74.9(23)

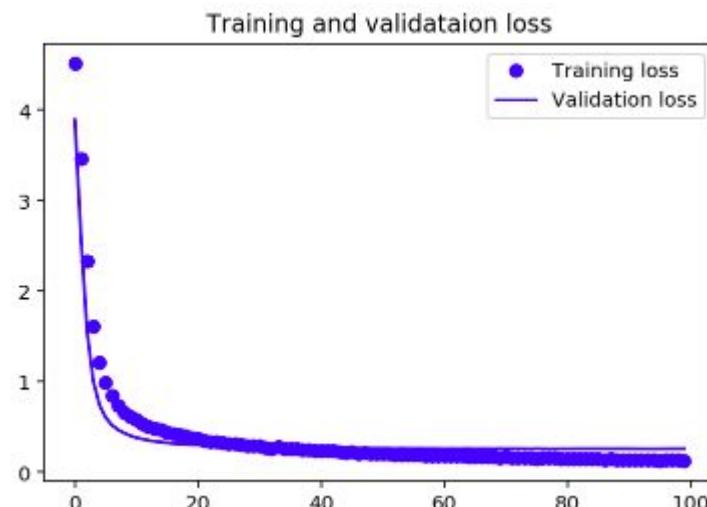
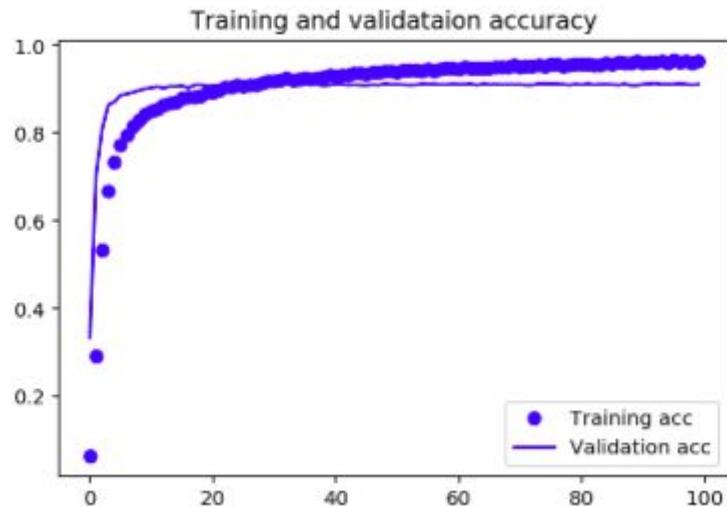
RESNET50

DataPrep	Pooling	Dropout	Dense	Activation	Optimizer	Batch Size	epoch	early stop	Val-Acc(epoch n)	Train-Acc(epoch n)	Val_loss
preprocess	avg	-	2: [256, 120]	[relu, softmax]	adam	100	100	10 acc	67.6(11)	59.7(13)	
preprocess	avg	-	2: [1024, 120]	[relu, softmax]	adam	100	100	10 acc	74.5(47)	83.0(42)	Overfit
preprocess	avg	-	1:[120]	[softmax]	adam	100	100	10 acc	80.0(51)	89.9(59)	Slightly Overfit

INCEPTION_V3 + RESNET152

DataPrep	Pooling	Dropout	Dense	Activation	Optimizer	Batch Size	epoch	early stop	Val-Acc	Train-Acc
preprocess	avg	0.5	2: [256, 120]	[relu]	adam	128	100	-	91.15	96.34

INCEPTION_V3 + RESNET152



Train loss: 0.1297, acc: 96.34%

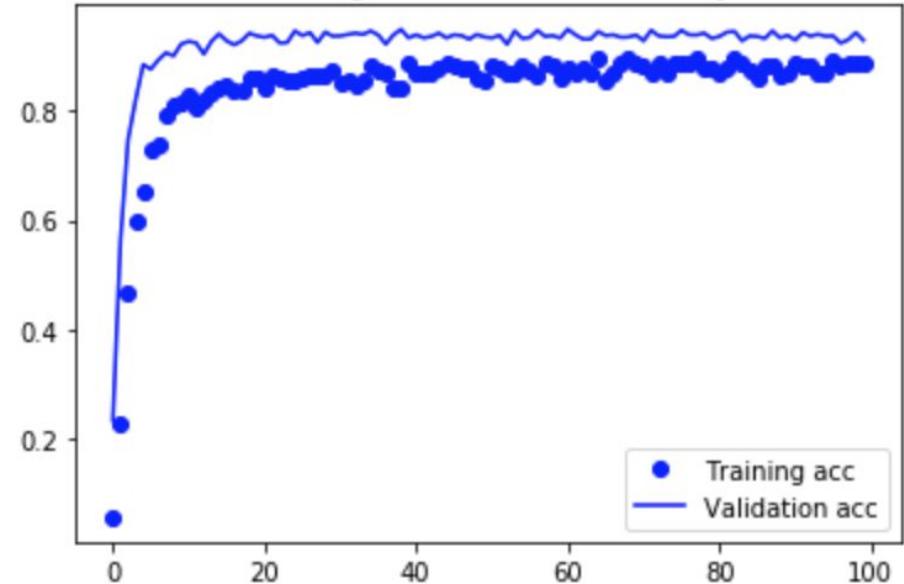
Validation loss: 0.2592, validation accuracy: 91.15%

INCEPTION_RESNET_V2

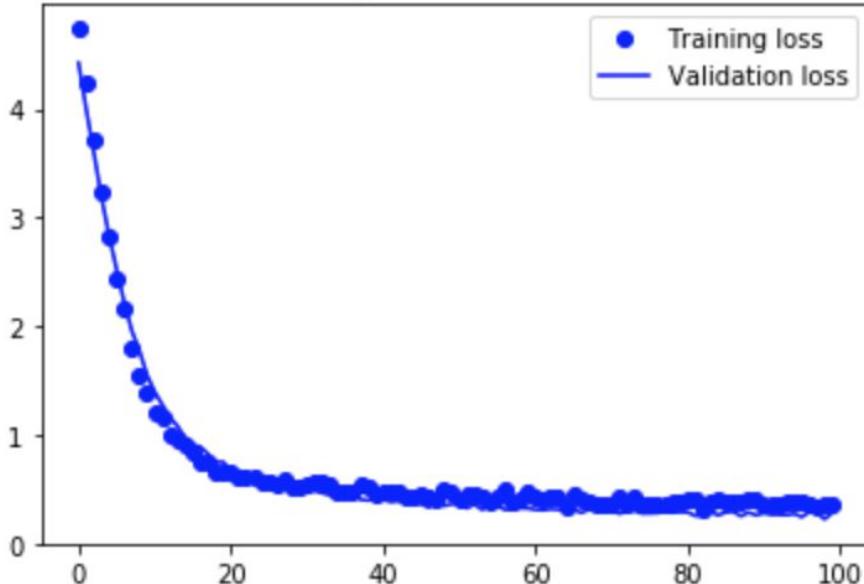
DataPrep	Freeze Layer	Pooling	Dropout	Dense	Activation	Optimizer	Val-Acc(epoch n)	Train-Acc(epoch n)	Val_loss	Train_loss
preprocess	all conv	avg		1:[120]	[softmax]	adam	92.3(15)	80.9(22)		
preprocess	all conv	avg		2: [256, 120]	[relu, softmax]	adam	92.0(31)	81.8(41)		
preprocess	all conv	avg		1:[120]	[softmax]	adam	94.6(26)	88.0(20)	0.42 / 36	0.49 / 35
preprocess	all conv	avg	0.5	1:[120]	[softmax]	adam	92.804 / 38	80.3 / 53	0.255 / 44	0.705 / 53
preprocess	[:-4]	avg	0.5	1:[120]	[softmax]	adam	92.6 / 25	79.8 / 37	0.269 / 40	0.707 / 45
preprocess	all conv	avg	0, 0.5	2: [1024, 120]	[relu, softmax]	adam	92.698 / 19	77.3 / 27	0.275 / 19	0.858 / 29
preprocess	all conv	max		2: [256, 120]	[relu, softmax]	adam	90.0 / 23	0.75 / 23	0.38 / 19	0.97 / 28
preprocess	all conv	avg		2: [1024, 120]	[relu, softmax]	adam	92.4 / 18	79.6 / 24	0.257 / 18	0.706 / 27
preprocess	all conv	avg		1:[120]	[softmax]	SGD	93.016 / 15	82.9 / 29	0.253 / 25	0.609 / 29
preprocess	all conv	avg	0.5	1:[120]	[softmax]	SGD	91.5 / 17	76.868 / 26	0.307 / 16	0.912 / 24
preprocess	all conv	avg		2: [1024, 120]	[relu, softmax]	SGD	90.7 / 13	76.049 / 16	0.32 / 13	0.853 / 16
preprocess	all conv	avg		1:[120]	[softmax]	adam	94.921 / 61	89.8 / 83	0.234 / 99	0.332 / 83
preprocess	all conv	avg		1:[120]	[softmax]	SGD	95.4 / 29	89.3 / 56	0.212 / 58	0.369 / 55
preprocess	all conv	avg		1:[120]	[softmax]	adam	94.2(22)	86.0(32)		

INCEPTION_RESNET_V2

Training and validation accuracy



Training and validation loss



TUNE LEARNING RATE USING RANDOM SEARCH

For SGD

	index	lr	acc	loss
0	2	0.252501	0.905820	1.076255
1	7	0.311364	0.902000	0.807524
2	3	0.384973	0.890000	0.663286
3	13	0.862344	0.886772	0.516855
4	12	0.202459	0.882540	1.414035
5	11	0.236449	0.880000	1.145891
6	8	0.971760	0.865000	0.887109
7	10	0.120291	0.828571	2.416401
8	1	0.123468	0.797884	2.419945
9	9	0.101897	0.794709	2.717516
10	4	0.054234	0.624000	3.626321
11	5	0.043199	0.475000	3.915030
12	0	0.043720	0.426455	3.942038
13	14	0.033032	0.384000	4.067627
14	6	0.030497	0.351000	4.145200

- first search lr in 10^{**-6} to 1, locate a better range from 0.01 to 1;
- second round search lr in 0.01-1, locate a better range from 0.1 to 1;

TUNE LEARNING RATE USING RANDOM SEARCH

For Momentum

	lr	second_para	acc	loss
0	[0.15293970003731822]	[0.9070221451794342]	0.902645	0.345246
1	[0.10078947191219727]	[0.942755736365718]	0.901587	0.356301
2	[0.12561562032003915]	[0.905256083791415]	0.901587	0.350621
3	[0.15293970003731822]	[0.9269409014648623]	0.9	0.384474
4	[0.1269207092444625]	[0.9026623029861273]	0.899	0.311226
5	[0.13226080533359527]	[0.9109932065258529]	0.898413	0.336487
6	[0.17029345260565862]	[0.9026623029861273]	0.898	0.334889
7	[0.13226080533359527]	[0.9081767447192606]	0.897	0.348427
8	[0.1947343991834089]	[0.983773707863622]	0.888	0.54189
9	[0.15293970003731822]	[0.9375316195214913]	0.886	0.4486
10	[0.17029345260565862]	[0.9525401893628124]	0.886	0.448851
11	[0.1798208780380528]	[0.9246230851786708]	0.885714	0.44077
12	[0.14544352811896316]	[0.9375316195214913]	0.885	0.406907
13	[0.18026493337089283]	[0.9884335691693487]	0.881	0.585928
14	[0.15293970003731822]	[0.9525401893628124]	0.880423	0.495603

- Learning rate between 0.1 to 0.2;
- Then find the best combination of momentum and initial learning rate, which is lr=0.15, momentum=0.9

Visualize Predictions

LABEL: boston LABEL: dingo LABEL: pekine LABEL: bluetic LABEL: golder LABEL: bedlin LABEL: borzoi LABEL: basenj LABEL: scottis LABEL: shetland_sheepdog (0.92)



LABEL: walker LABEL: malte: LABEL: norfolk LABEL: africar LABEL: wire-h LABEL: redbor LABEL: lakela LABEL: boxer LABEL: doberri LABEL: otterhound (1.00)



LABEL: stande LABEL: irish_v LABEL: black-: LABEL: cairn LABEL: affenp LABEL: labrad LABEL: ibizan LABEL: englisl LABEL: weima LABEL: giant_schnauzer (0.79)



LABEL: groene LABEL: dhole LABEL: toy_pc LABEL: border LABEL: tibetar LABEL: norwe LABEL: shih-t LABEL: irish_t LABEL: kuvas: LABEL: german_shepherd (0.88)



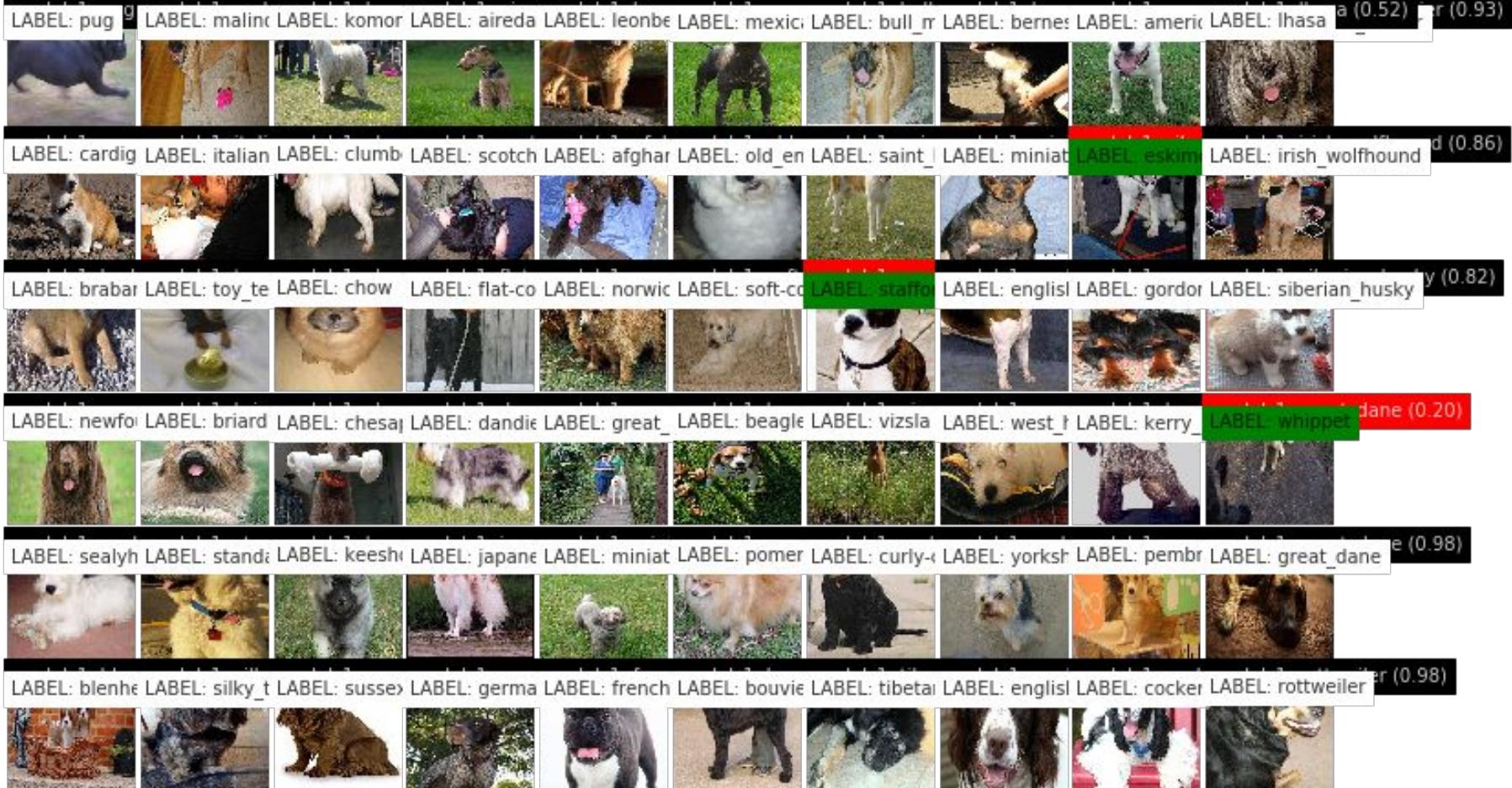
LABEL: greate LABEL: bassel LABEL: austra LABEL: schipp LABEL: rhode: LABEL: irish_s LABEL: appen LABEL: bloodl LABEL: samoy LABEL: miniature_schnauzer (0.80)



LABEL: brittar LABEL: kelpie LABEL: papillc LABEL: border LABEL: entleb LABEL: collie LABEL: malar LABEL: welsh LABEL: chihu LABEL: saluki (0.98)



Visualize Predictions



CONCLUSION

- **Best Model:** Inception_Resnet_V2
- **Val Accuracy:** 95%
- **Hyperparameter Takeaways:**
 - Larger Resolution
 - Adam optimizer(easy to use)
 - SGD (fast but hard to find)

THANK YOU!

