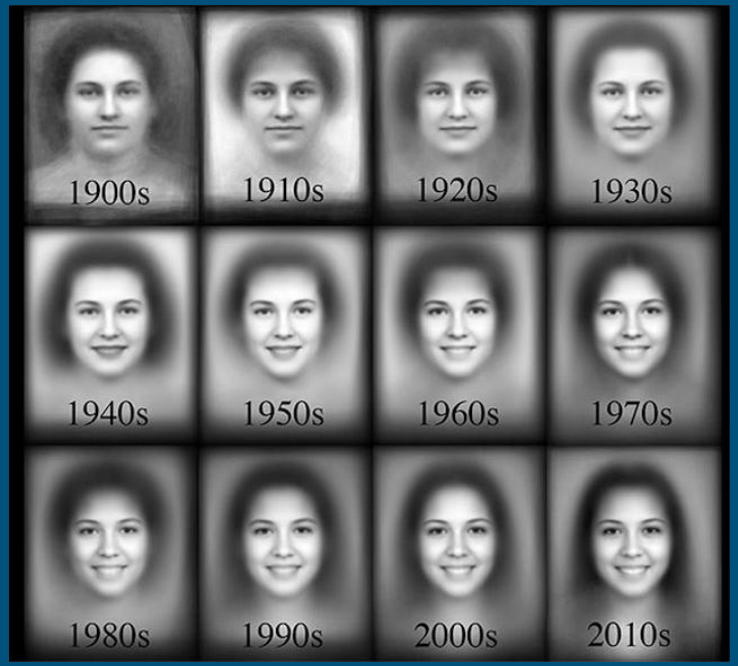
Yearbook dating and Geolocation prediction with CNNs

Chandana Amanchi
Pandian Raju

Yearbook dating



Classification using CNNs

- AlexNet
- VGG
- ResNet
- DenseNet
- **Ensemble**

Image Credit: https://hyperallergic.com/261201/the-rise-of-the-20th-century-yearbook-smile/

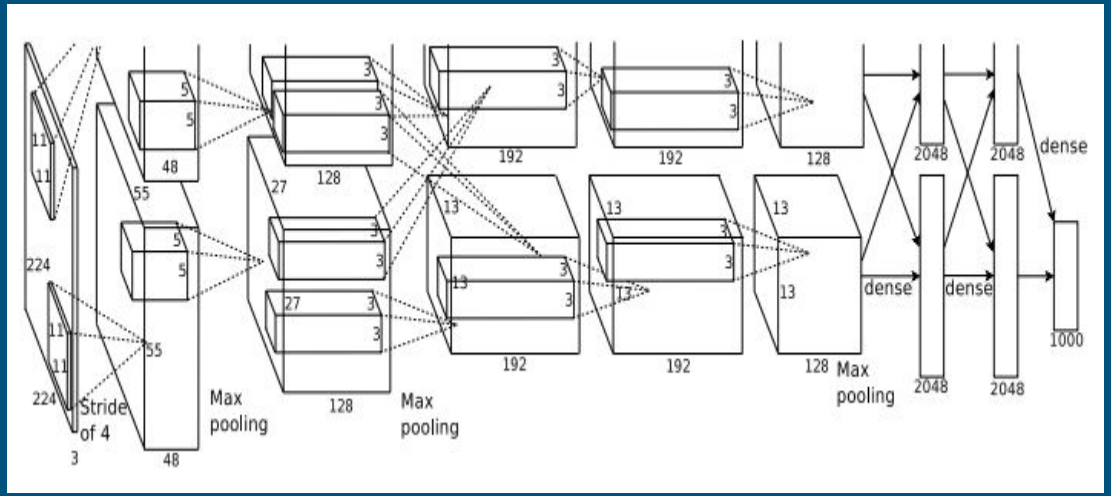
Geolocation prediction

Prediction as regression

Prediction as classification

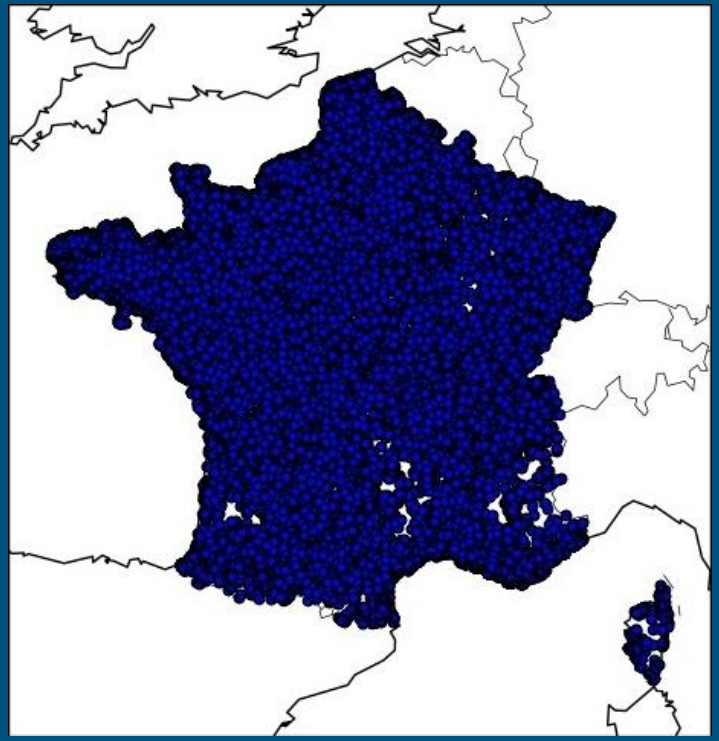
Image Credit: http://www.philkr.net/cs395t/project1/

Geolocation as regression



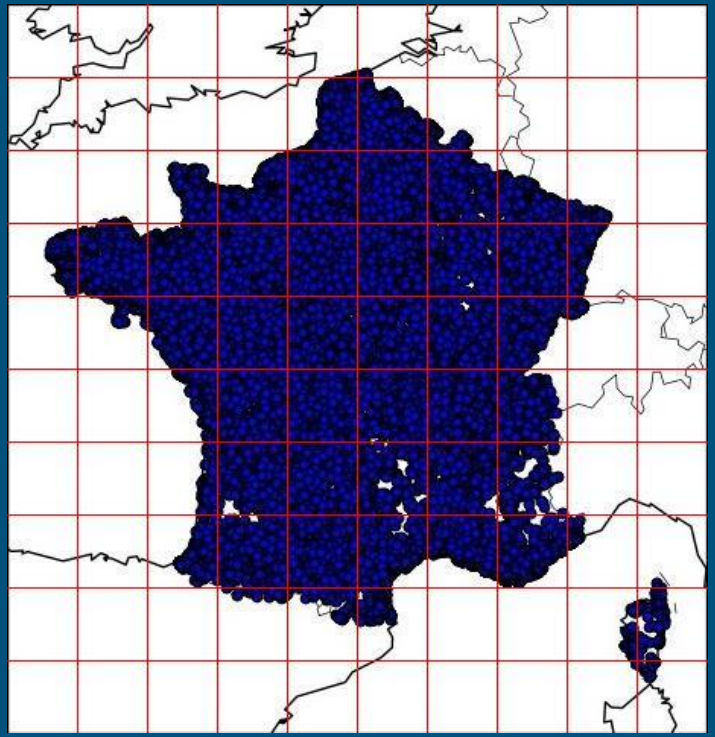
- Four convolutional layers
- Two fully connected layers
- Final dense layer of dimension 2

Geolocation as classification



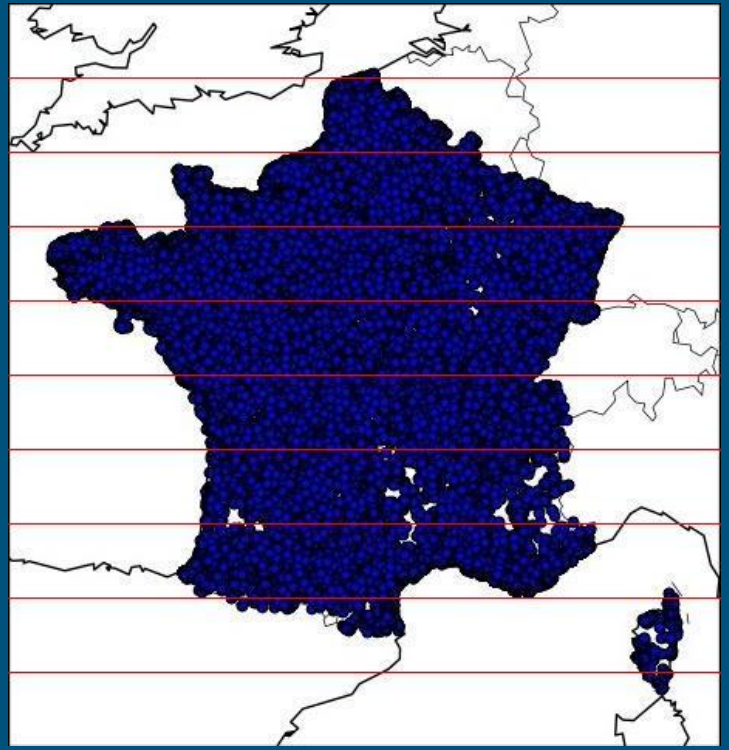
Given data in x-y coordinates

Geolocation as classification



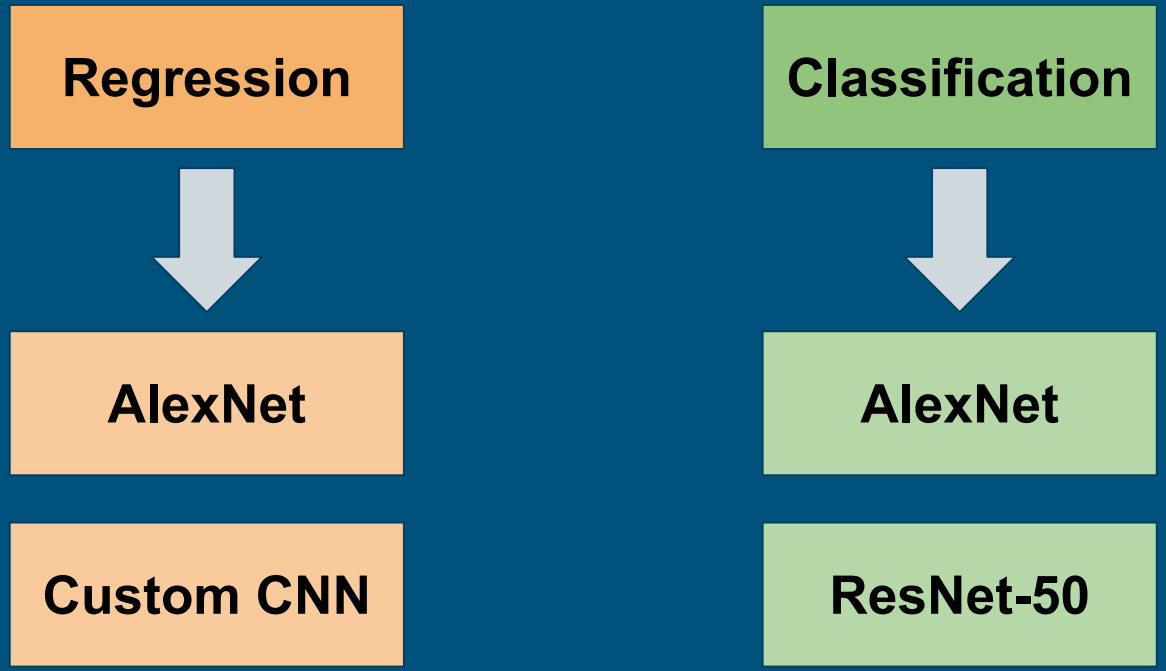
- Divide as 10x10 grid
- Side of 1 box = 160 km
- Misclassification penalty: 160 km

Geolocation as classification



- Smaller grid for more dense data
- For eg, each box has fixed height, but variable width

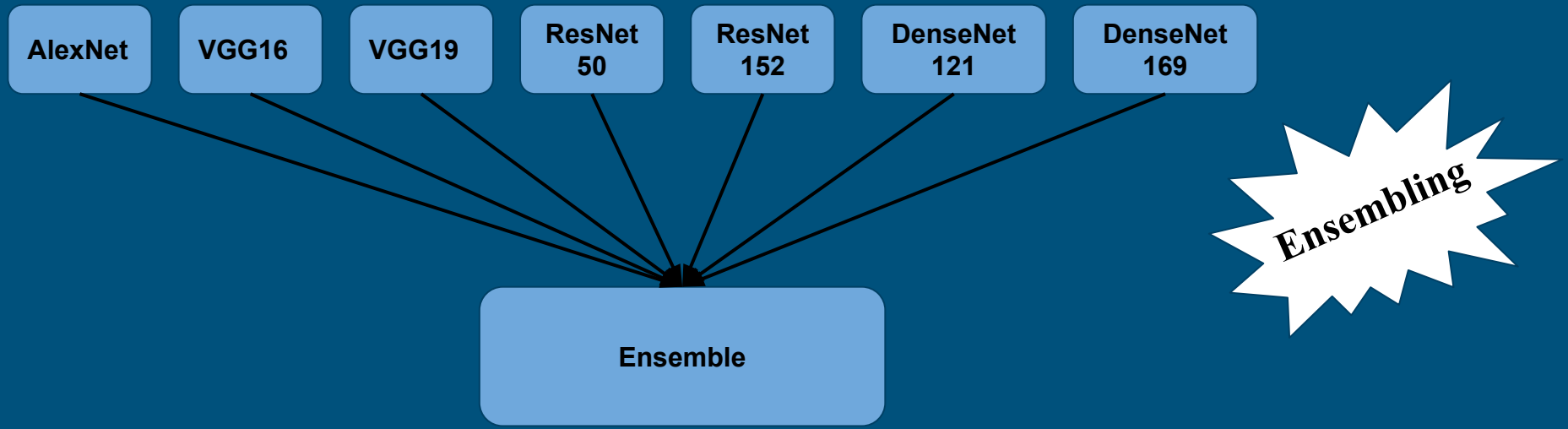
Geolocation as classification



Other techniques

- Use ImageNet pre-trained weights with smaller learning rate
- Freezed the initial few convolutional layers of the network

What(we think!) sets the team apart



Challenges

- **Training deeper networks**
- Memory constraints
- Tuning hyper-parameters

Image Credit: https://giphy.com/gifs/sloth-zootopia-meet-4pGwaW3nuRD1e

Challenges

- Training deeper networks
- **Memory constraints**
- Tuning hyper-parameters

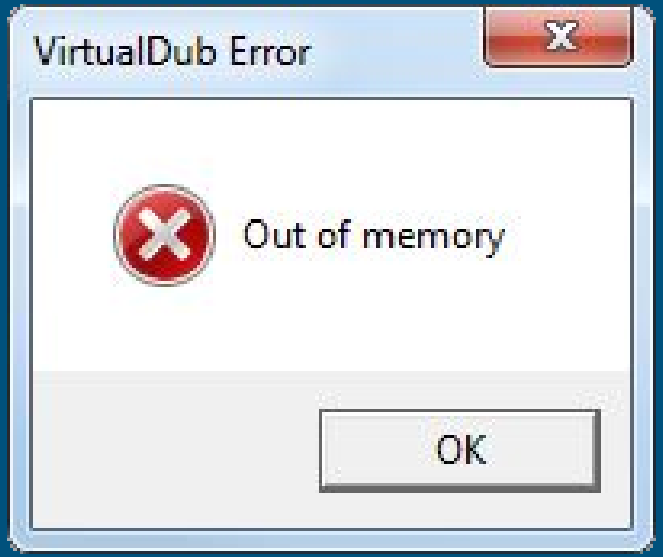


Image Credit: https://stackoverflow.com/questions/18720919/avisynth-out-of-memory-error-100s-of-image-overlays

Challenges

- Training deeper networks
- Memory constraints
- **Tuning hyper-parameters**

Image Credit: http://marxrepair.com/repair-services/diesel-mechanic-repair-services/

Tuning hyper-parameters

Run	Architecture	Comments	Validation L1
1	AlexNet	AlexNet 10 epochs batch size 128 mse loss	29.19
2	AlexNet	AlexNet 20 epochs batch size 128 mse loss	19.36
3	AlexNet	AlexNet 20 epochs batch size 256 mse loss	31.97
4	AlexNet	AlexNet 40 epochs batch size 128 mse loss	19.87
5	-	-	-
6	AlexNet	AlexNet 20 epochs batch size 128 l1-custom loss	38.04
7	AlexNet	AlexNet 40 epochs batch size 64 l1-custom loss	36.57
8	AlexNet	AlexNet 20 epochs batch size 128 mse loss small learning rate	27.64
9	VGG16	VGG16 5 epochs batch size 32 cc loss freezing initial layers	6.98
10	VGG16	VGG16 10 epochs batch size 128 cc loss freezing initial layers	-
11	DenseNet169	DenseNet 5 epochs batch size 16 cc loss	-
12	VGG16	VGG16 20 epochs batch size 128 cc loss freezing initial layers (10 layers)	6.4
13	DenseNet169	DenseNet 10 epochs batch size 16 cc loss	5.02
14	VGG16	VGG16 20 epochs batch size 64 cc loss without freezing initial layers	6.4
15	AlexNet	AlexNet 50 epochs batch size 128 cc loss without freezing initial layers	6.52
16	AlexNet	AlexNet 50 epochs batch size 128 cc loss freezing initial layers	5.82
17	AlexNet	AlexNet 50 epochs batch size 128 l1 loss freezing initial layers	28.9
18	AlexNet	AlexNet 50 epochs batch size 128 mse loss freezing initial layers	16.13
19	ResNet50	Resnet50 5 epochs batch size 128 cc loss freezing initial layers	10.67817928
20	ResNet50	Resnet50 15 epochs batch size 128 cc loss freezing initial layers	9.149381204
21	AlexNet	Alexnet 15 epochs batch size 128 cc loss freezing initial layers	6.208225195
22	AlexNet	AlexNet 50 epochs batch size 128 cc loss freezing initial layers	5.753443801
23	AlexNet	AlexNet 150 epochs batch size 256 mse loss freezing initial layers	16.6
24	AlexNet	AlexNet 150 epochs batch size 512 mse loss freezing initial layers	19.9
25	AlexNet	AlexNet 100 epochs batch size 512 cc loss freezing initial layers	5.5
26	VGG16	VGG16 20 epochs batch size 256 cc loss freezing initial layers	-
27	VGG16	VGG16 20 epochs batch size 512 cc loss freezing initial layers	-
28	AlexNet	AlexNet 150 epochs batch size 512 mse loss freezing initial layers (higher learning rate)	?
29	Ensemble	Ensembling: vgg16:checkpoint12-5-11-12.h5,densenet169:checkpoint13-0-1-2-3.h5	4.89
30	Ensemble	Ensembling: vgg16:checkpoint12-5-11-12.h5,densenet169:checkpoint13-0-1-2-3-4-5.h5	4.77
31	Ensemble	Ensembling: vgg16:checkpoint12-5-11-12.h5,densenet169:checkpoint13-0-1-2-3-4-5.h5,alexnet:checkpoint16-38.h5	-
32	AlexNet	AlexNet 150 epochs batch size 512 cc loss freezing initial layers (higher learning rate)	-
33	AlexNet	AlexNet 100 epochs batch size 128 cc loss freezing initial layers	5.47
34	AlexNet	AlexNet 100 epochs batch size 256 cc loss freezing initial layers	5.38
35	AlexNet	AlexNet 100 epochs batch size 512 cc loss freezing initial layers	5.57
36	AlexNet	AlexNet 150 epochs batch size 512 mse loss freezing initial layers (higher learning rate)	
37	Ensemble	Ensembling: vgg16:checkpoint12-5-11-12.h5,densenet169:checkpoint13-0-1-2-3-4-5-6-7.h5,alexnet:checkpoint34-95-weights.h5	4.27
38	AlexNet	AlexNet 100 epochs batch size 128 l1 (wo abs) loss freezing initial layers	31.83
39	AlexNet	AlexNet 100 epochs batch size 512 cc loss freezing initial layers (lr 1e-4)	5.68
40	ResNet50	Resnet50 16 epochs batch size 128 cc loss freezing initial layers (lr 10^-2)	8.36
41	ResNet50	Resnet50 mse (lr 10^-2) - 6 epochs	39.25
42	ResNet50	Resnet50 l1 (lr 10^-2) - 6 epochs	30.11
43	ResNet152	resnet152_5e_128b_cc_sgd_fi_lr10-3	11.47
44	ResNet152	resnet152_5e_128b_cc_sgd_fi_lr10-2	10.11
45	ResNet50	resnet50 lr 10^-2 training only the last layer	8.35

46	Kaggle	stview 10^-3	309.78
47	Kaggle	stview 10^-2	345.45
48	VGG16	VGG16 20 epochs batch size 128 cc loss freezing initial layers (24 layers)	6.3
49	VGG19	VGG19 20 epochs batch size 128 cc loss freezing initial layers (19 layers)	6.03
50	DenseNet121	DenseNet121 10 epochs batch size 16 cc loss	6.06
51	DenseNet161	DenseNet161 10 epochs batch size 16 cc loss	-
52	AlexNet	AlexNet 100 epochs batch size 256 cc loss freezing initial layers (initial 20 layers)	5.3
53	ResNet50	resnet50 freeze 100 layers lr 10-3	6.48
54	ResNet50	resnet50 freeze 100 layers lr 10-2	6.17
55	ResNet152	resnet152 freeze 500 layers 10-3	Timed out - 16h
56	ResNet152	resnet152 freeze 500 layers 10-2	Timed out - 16h
57	Kaggle	stview 10^-1	352
58	AlexNet	Stview - 100 epochs batch size 256 cc loss freezing initial layers (20 layers) - 20 x 20	358
59	AlexNet	Stview - 100 epochs batch size 512 cc loss freezing initial layers (20 layers) - 20 x 20	362
60	AlexNet	Stview - 100 epochs batch size 512 cc loss freezing initial layers (20 layers) - 20 x 20	360
61	AlexNet	Stview - 100 epochs batch size 512 cc loss freezing initial layers (20 layers) - 20 x 20	367
62	AlexNet	Stview - 100 epochs batch size 1024 cc loss freezing initial layers (20 layers) - 20 x 20	364
63	AlexNet	Stview - 100 epochs batch size 512 cc loss freezing initial layers (20 layers) - 30 x 30	366
64	AlexNet	Stview - 100 epochs batch size 512 cc loss freezing initial layers (20 layers) - 40 x 40	368
65	AlexNet	Stview - 100 epochs batch size 512 cc loss freezing initial layers (20 layers) - 50 x 50	368
66	Ensemble	Ensembling: vgg16:checkpoint12-5-11-12.h5,alexnet:checkpoint34-95-weights.h5,vgg19:checkpoint49-4-5.h5,densenet169:chec	4.1
67	AlexNet	Stview - 100 epochs batch size 512 cc loss freezing initial layers (20 layers) - 20 x 20 - lr 1e-2	367
68	AlexNet	Stview - 100 epochs batch size 512 cc loss freezing initial layers (20 layers) - 10 x 10	357
69	AlexNet Regressio	Stview	529
70	Resnet50	freeze 50 layers lr 10-2	5.78
71	Resnet50	end to end fine tuning lr 10-3	cuda memory er
72	Resnet152	Freeze 600 layers batch 32 lr 102	7.75
73	Keras Resnet50	Stview - batch size 128 - cc loss - sgd lr 1e-3 - 10 x 10	336
74	Keras Resnet50	Stview - batch size 128 - cc loss - adam lr 1e-4 - 10 x 10	322
75	Keras Resnet50	Stview - batch size 128 - cc loss - sgd lr 1e-3 - 20 x 20	353
76	Keras Resnet50	Stview - batch size 128 - cc loss - adam lr 1e-4 - 20 x 20	319
77	Keras Resnet50	Stview - batch size 128 - cc loss - sgd lr 1e-2 - 10 x 10	312
78	Keras Resnet50	Stview - batch size 128 - cc loss - adam lr 1e-3 - 10 x 10	305.2
79	kaggle	Stview 10-4	383
80	Resnet152	freeze 500 layers batch 64 lr 10-3	6.76
81	Ensemble	resnet50:resnet50checkpoint17-6-15.h5,vgg16:checkpoint12-5-11-12.h5,alexnet:checkpoint34-95-weights.h5,vgg19:checkpoint4	4.02
82	Keras Resnet50	Stview - batch size 128 - cc loss - adam lr 1e-4 - 50 x 50	338
83	Keras Resnet50	Yearbook batch size 128 - cc loss - sgd 1e-2	-
84	Keras Resnet50	Yearbook batch size 128 - cc loss - sgd 1e-3	-
85	Keras Resnet50	Yearbook batch size 128 - cc loss - adam 1e-3	7.5
86	Keras Resnet50	Yearbook batch size 128 - cc loss - adam 1e-4	-
87	Ensemble	resnet50:resnet50checkpoint18-5.h5,vgg16:checkpoint12-5-11-12.h5,alexnet:checkpoint34-95-weights.h5,vgg19:checkpoint49-4-5.h5,densenet169:	
88	Ensemble	kaggle:kagglecheckpoint1-4-9-14-19.h5,keras_resnet50:checkpoint78-0-1-2.h5	265.52
89	Ensemble	resnet50:resnet50checkpoint18-5.h5,vgg16:checkpoint12-5-11-12.h5,alexnet:checkpoint34-95-weights.h5,vgg19:checkpoint49-4-5.h5,densenet169:	

Fun

- Custom loss function for yearbook dating

Fun

- Custom loss function for yearbook dating
- Ensembling over labels versus ensembling over softmax probabilities

Fun

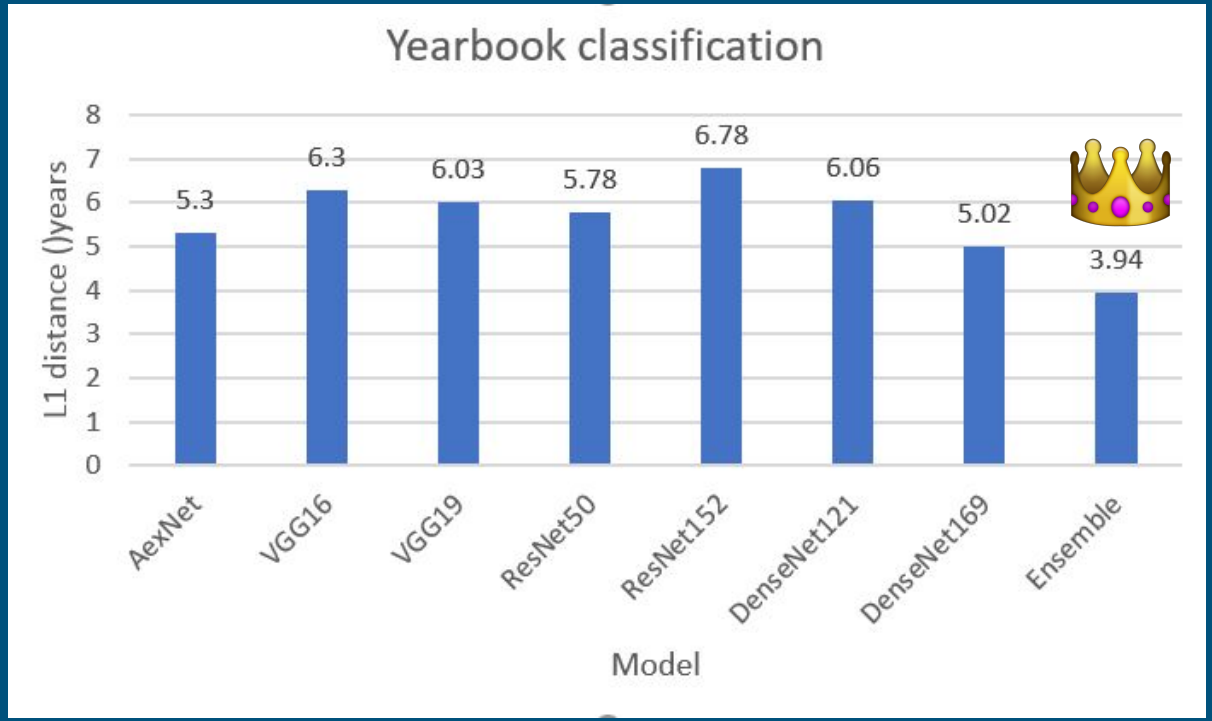
- Custom loss function for yearbook dating
- Ensembling over labels versus ensembling over softmax probabilities
- Exhaustive combinations of ensemble models

[AlexNet, VGG16, VGG19, Dense169, Dense121, ResNet50, ResNet169]

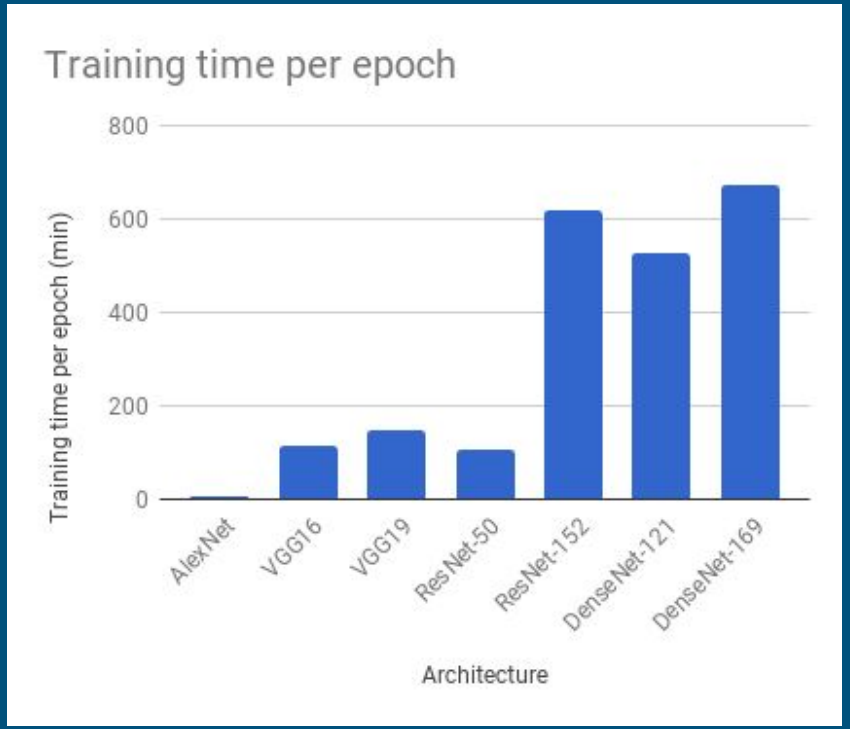
Number of combinations? 128

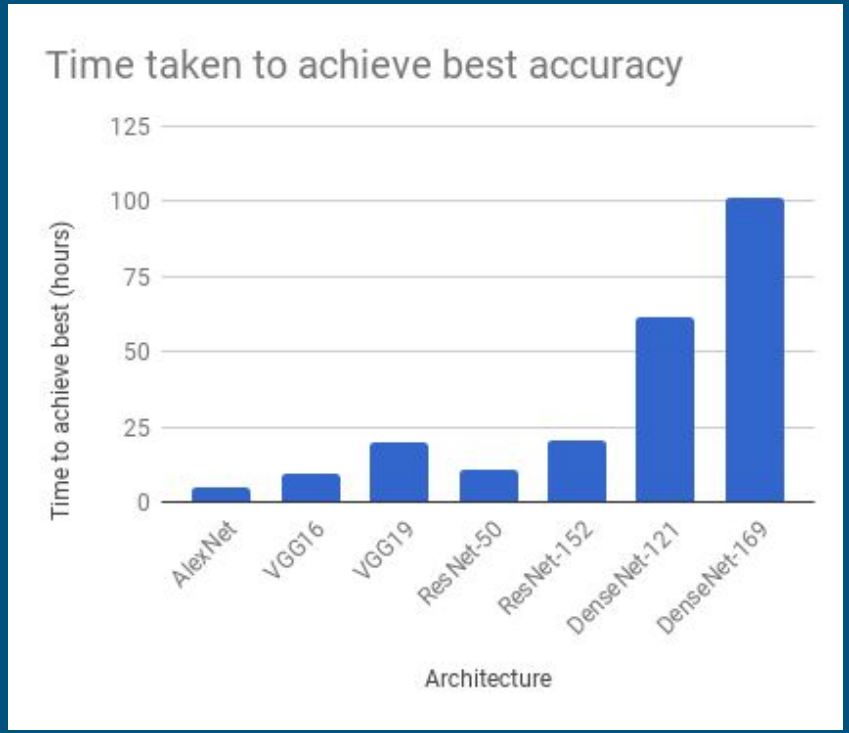
How could we have done this efficiently?

Evaluation - Yearbook

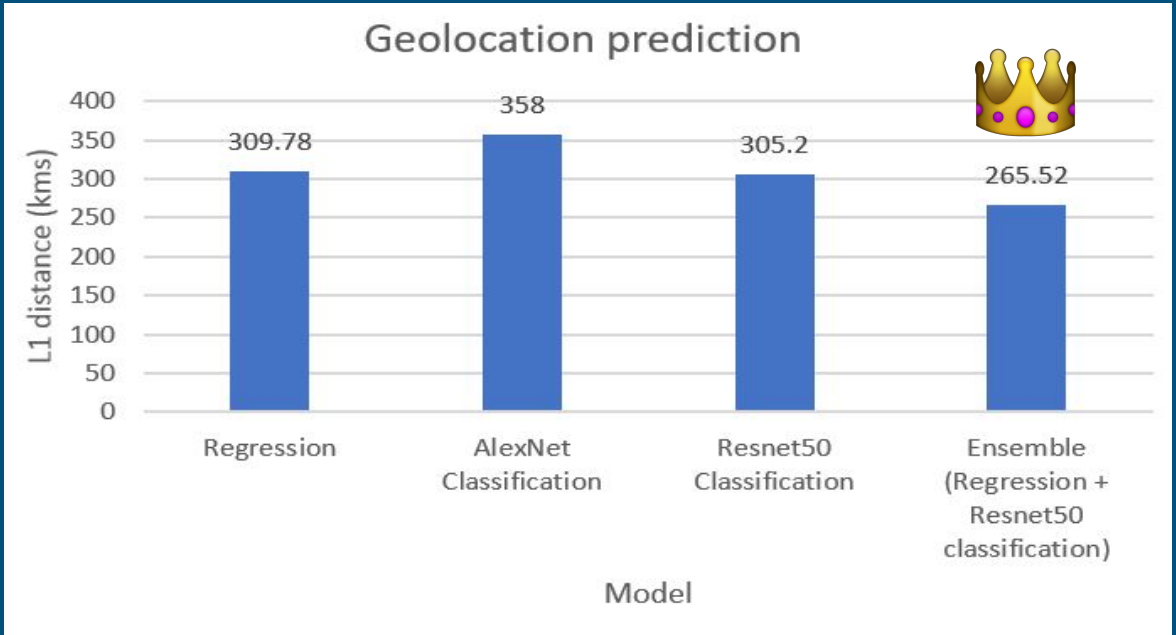


Training time - Yearbook





Evaluation - Geolocation



Could have achieved higher accuracy with better parameter tuning, better classification logic, time and infinite resources (:-P)

Questions?