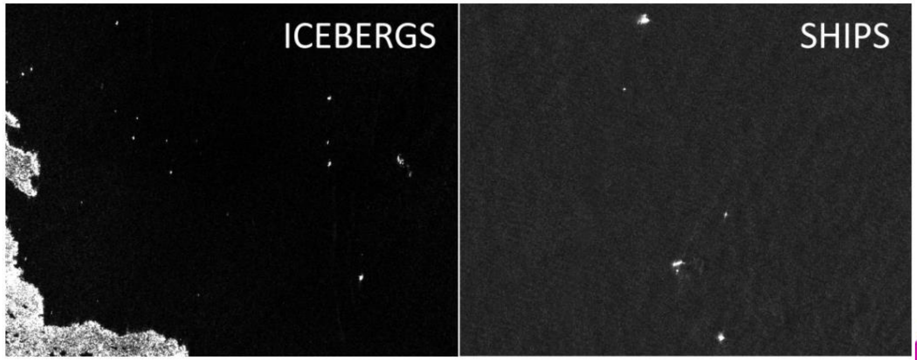
AIDC

INTEL AI DEVCON 2018

WINNING SOLUTION TO KAGGLE'S SHIP VS ICEBERG COMPETITION

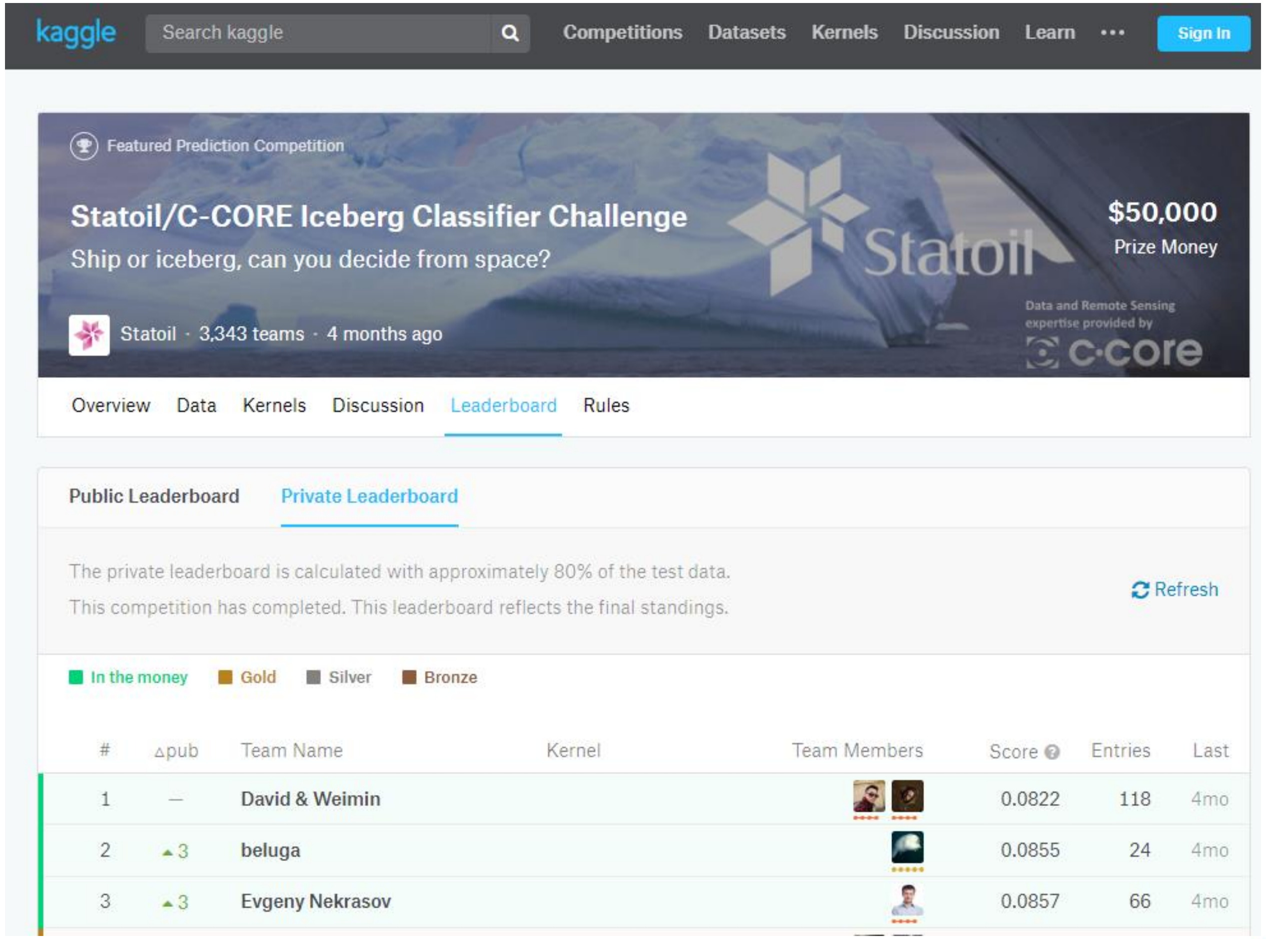
David Austin & Weimin Wang

5/23/2018



SHIPS VS ICEBERGS COMPETITION

- AI Competitions
- General workflow
- Specific solution workflow
- Tips for competitions



AI COMPETITIONS

	Real World	Competitions
Establish business problem	Yes	No
Collect data	Yes	No
Deployment Considerations	Yes	No
Complexity vs Accuracy	Yes	No
Algorithm constraints	Yes	Maybe
Problem target value	Yes	Yes

Adapted from Coursera's "How to Win a Data Science Competition"

COMPETITION SPONSORS

kaggle

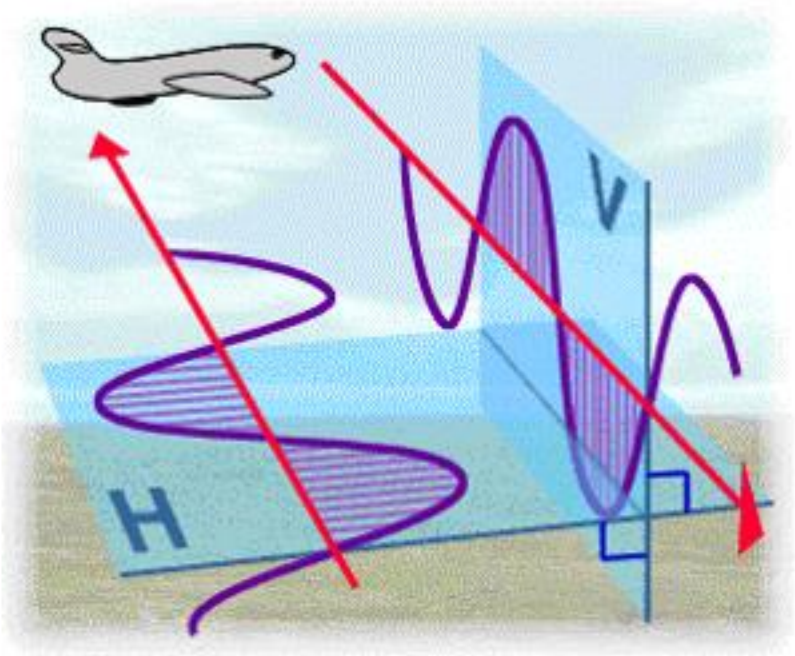
- Platform for data science competitions.
- Hosts competitions by providing community, data, and prize.
- Data science community collaborates and competes to develop solutions.
- Active users: 895k (90% growth), 41 competitions in 2017

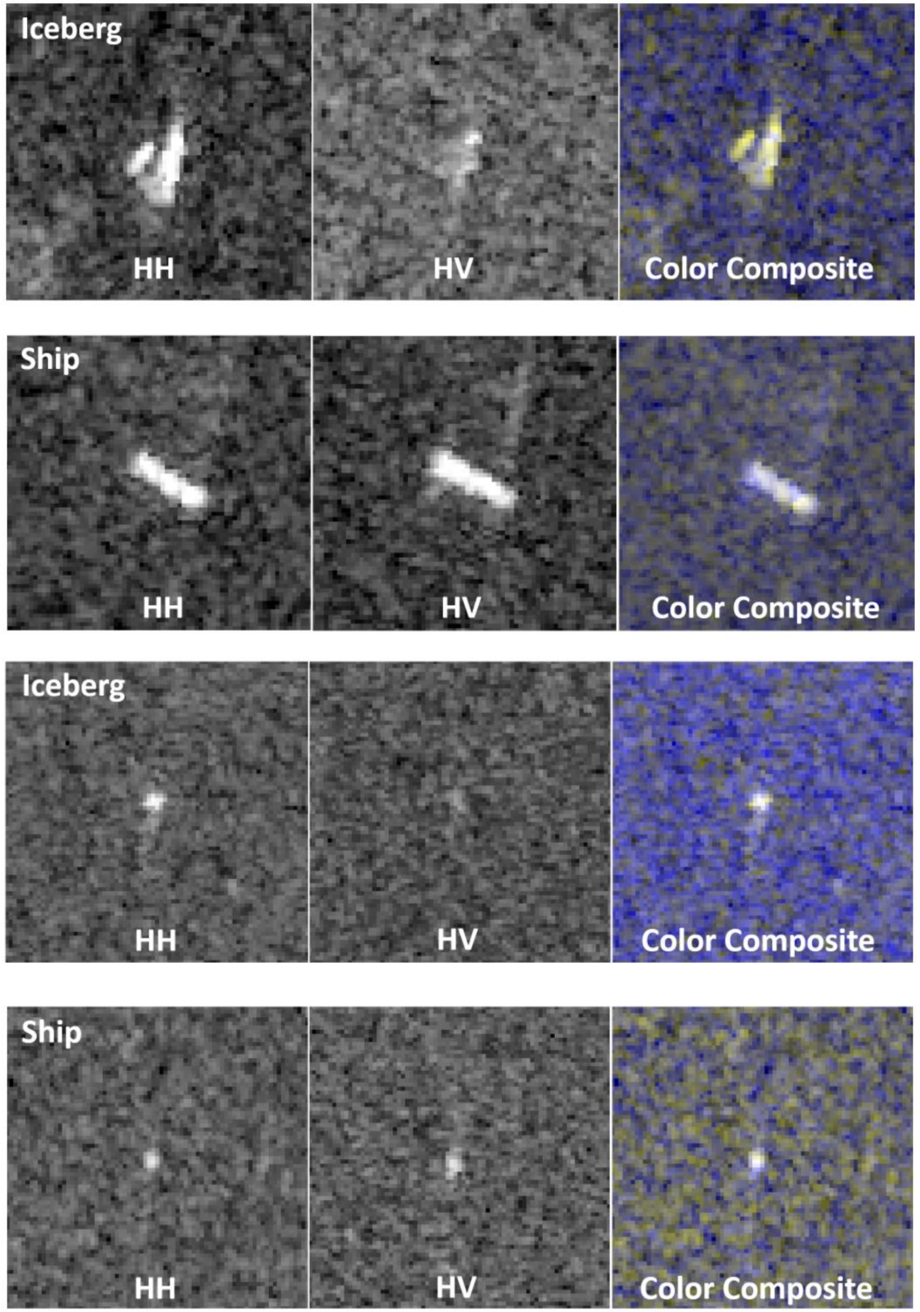
Statoil

- Norwegian based energy company, presence in 20 countries.
- Focused on energy exploration, development, and production of oil, gas, and wind power.
- 20.5k employees, listed on NYSE. $~60B in 2017 revenue.

THE PROBLEM

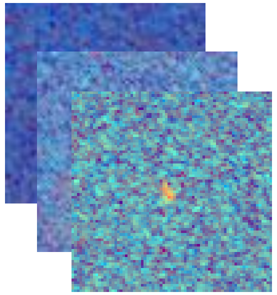
Develop an algorithm to classify ships vs icebergs using SAR satellite images



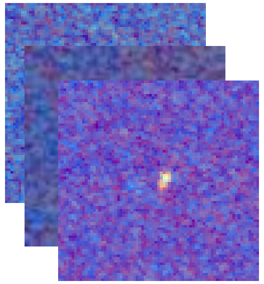


DATASET

icebergs



ships



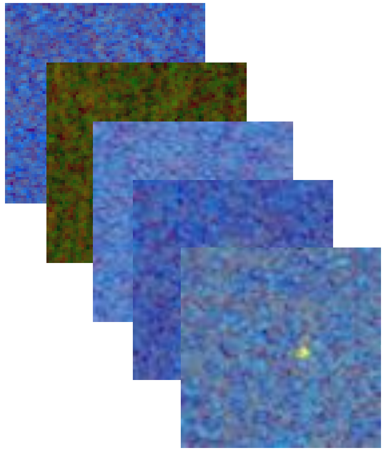
3425 images (75x75x2)

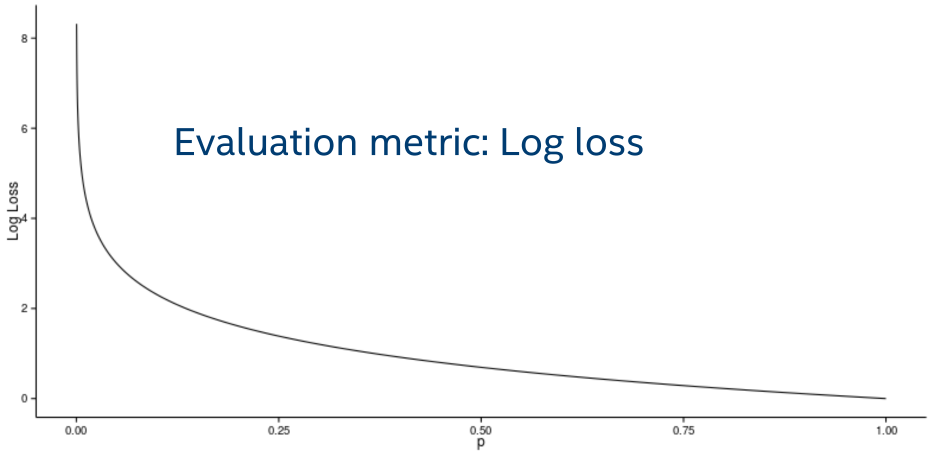
PREDICT

MODEL

INPUT

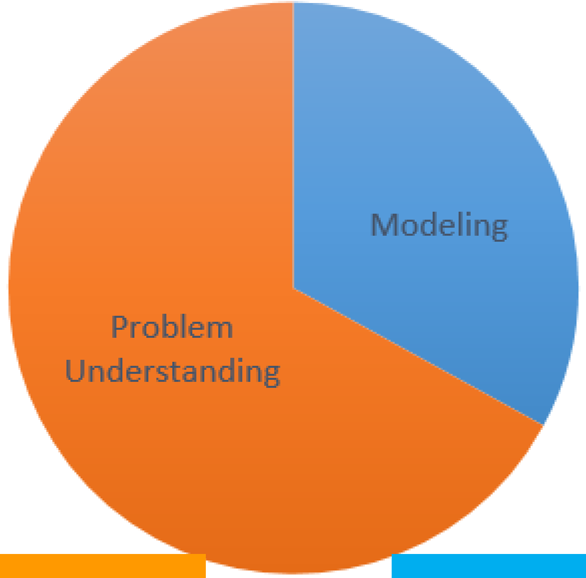
1604 images [(75,75,2), inc_angle]





Evaluation metric: Log loss

SOLUTION PIPELINE



PROBLEM UNDERSTANDING

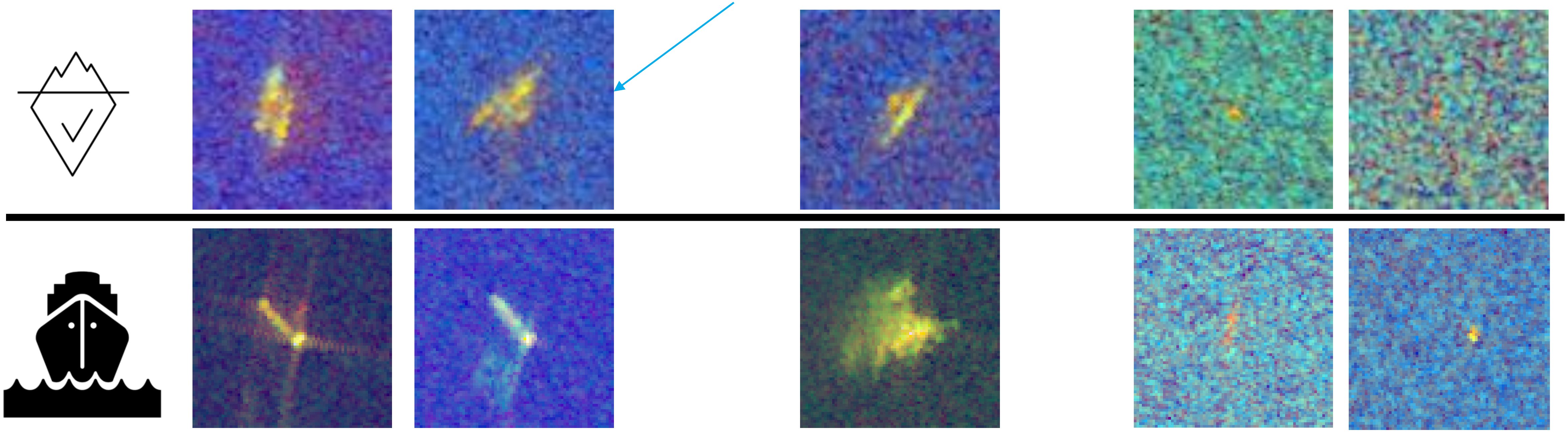
- EDA
- Model selection

MODELING

- Algo Eval
  - Diversity, diversity, diversity
- Ensembling
- Post processing

IMAGE EXPLORATION

Asymmetrical, larger volume, amorphous



Side lobes, diffraction spikes

Indistinguishable
Possible background bias

Conclusion from visual analysis: No clear consistent pattern that would guide specific design considerations.

IMAGE VISUALIZATIONS: GRAD-CAM

Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

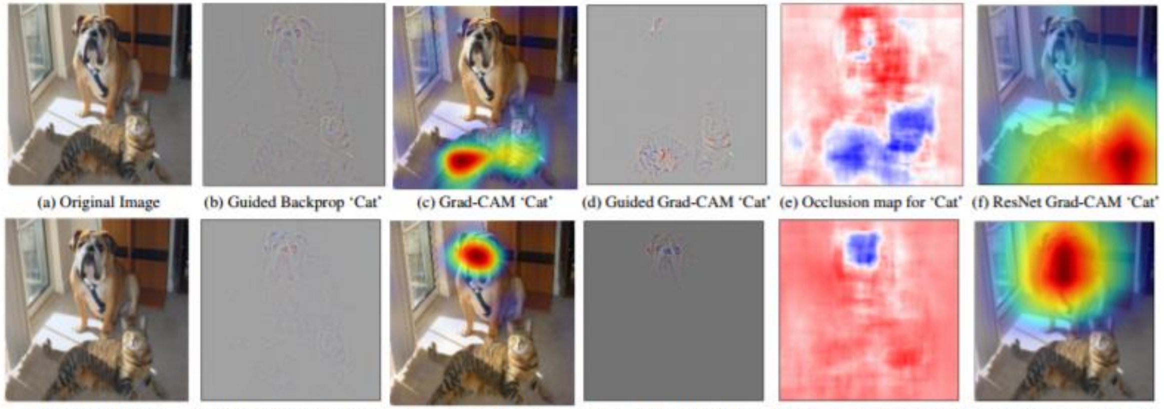
Ramprasaath R. Selvaraju[1] Michael Cogswell[2] Abhishek Das[2] Ramakrishna Vedantam[1]
Devi Parikh[2] Dhruv Batra[2]
[1]Virginia Tech, [2]Georgia Institute of Technology
{ram21, vrama91}@vt.edu
{cogswell, abhshkdz, vrama91, parikh, dbatra}@gatech.edu

Abstract

We propose a technique for producing 'visual explanations' for decisions from a large class of Convolutional Neural Network (CNN)-based models, making them more transparent. Our approach – Gradient-weighted Class Activation Mapping (Grad-CAM), uses the gradients of any target concept (say logits for 'dog' or even a caption), flowing into the final convolutional layer to produce a coarse localization

1. Introduction

Convolutional Neural Networks (CNNs) and other deep networks have enabled unprecedented breakthroughs in a variety of computer vision tasks, from image classification [27, 18] to object detection [16], semantic segmentation [31], image captioning [47, 7, 13, 23], and more recently, visual question answering [3, 15, 36, 41]. While these deep neural networks enable superior performance, their lack of

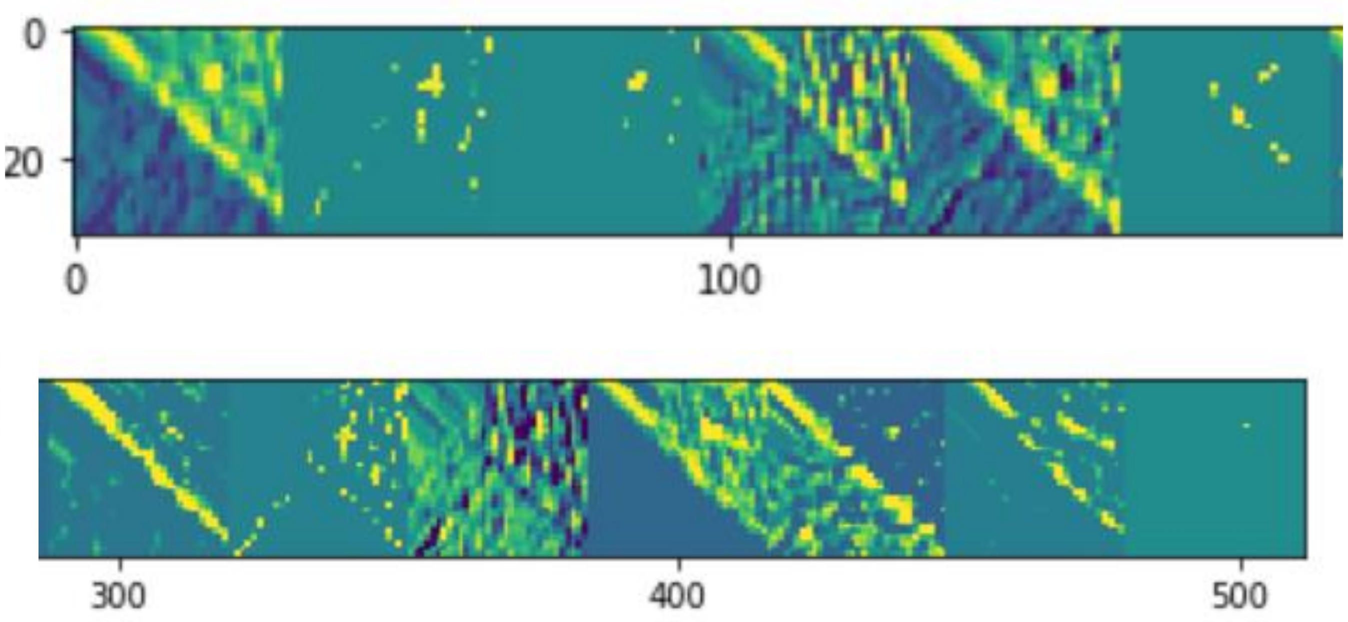


(a) Original Image (b) Guided Backprop 'Cat' (c) Grad-CAM 'Cat' (d) Guided Grad-CAM 'Cat' (e) Occlusion map for 'Cat' (f) ResNet Grad-CAM 'Cat'

(g) Original Image (h) Guided Backprop 'Dog' (i) Grad-CAM 'Dog' (j) Guided Grad-CAM 'Dog' (k) Occlusion map for 'Dog' (l) ResNet Grad-CAM 'Dog'

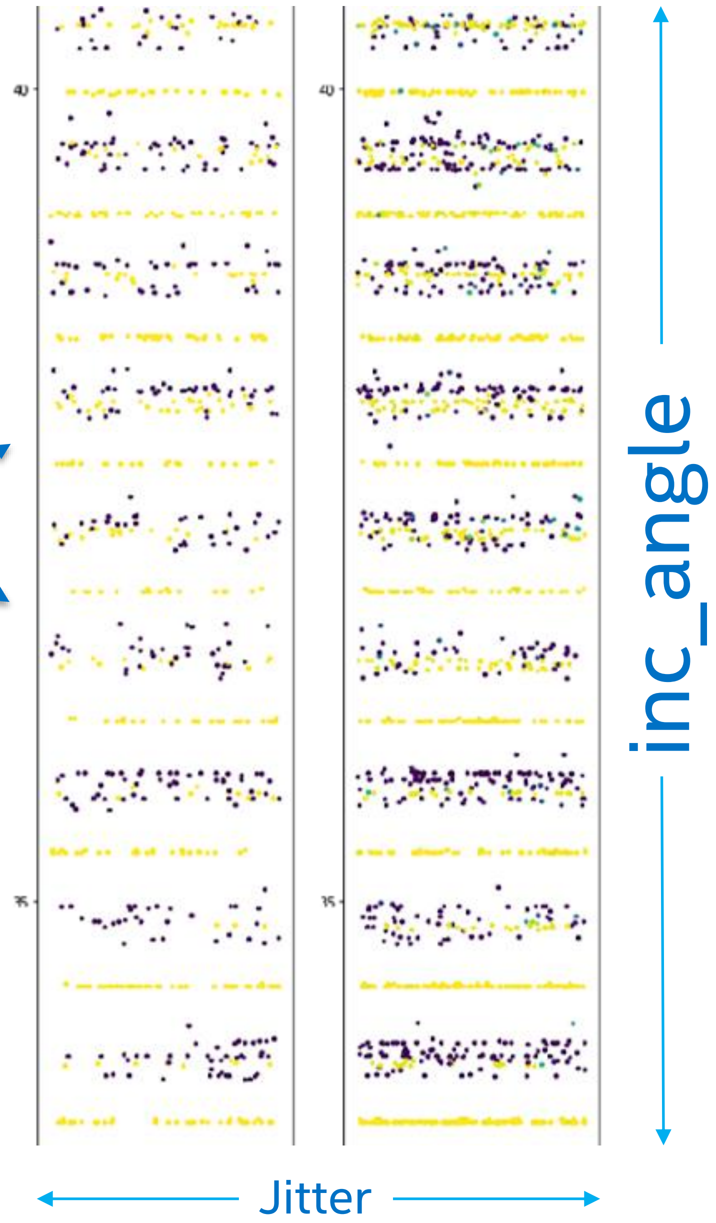
Figure 1: (a) Original image with a cat and a dog. (b-f) Support for the cat category according to various visualizations for VGG and ResNet. (b) Guided Backpropagation [46]: highlights all contributing features. (c, f) Grad-CAM (Ours): localizes class-discriminative regions, (d) Combining (b) and (c) gives Guided Grad-CAM, which gives high-resolution class-discriminative visualizations.Interestingly, the localizations achieved by our Grad-CAM technique, (c) are very similar to results from occlusion sensitivity (e), while being orders of magnitude cheaper to compute. (f, l) are Grad-CAM visualizations for ResNet-18 layer. Note that in (d, f, i, l), red regions corresponds to high score for class, while in (e, k), blue corresponds to evidence for the class. Figure best viewed in color.

Example intermediate convolution visualization



KEY FINDING

Clear alternating patterns of dense clusters of ships vs icebergs

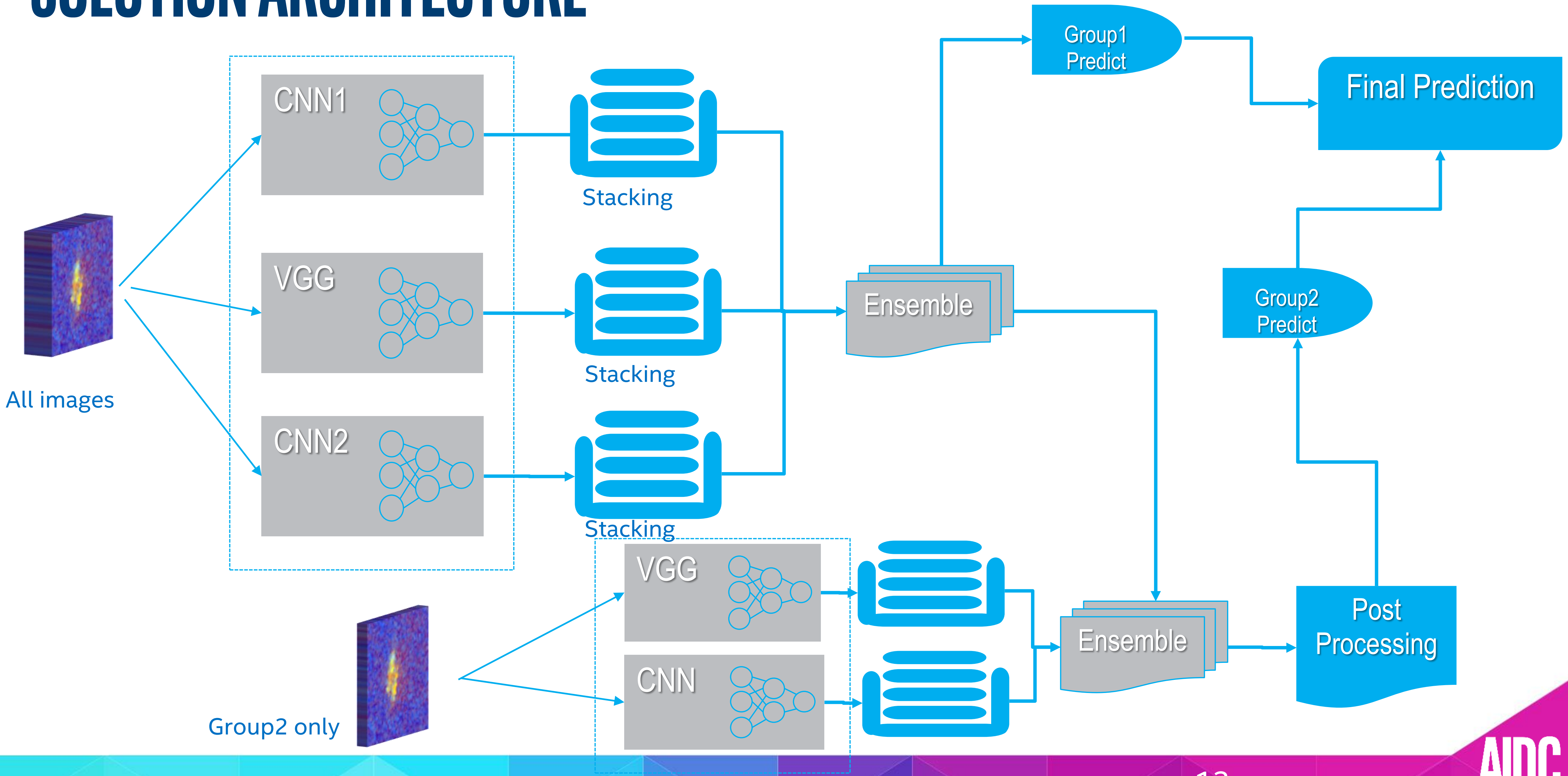


	angle	N Rows	N(test)	N(train)	Mean(is_iceberg, test)	Mean(is_iceberg, train)
1	34.4721	59	36	23	0.9020344	1
2	42.5591	51	34	17	0.90264152	0.94117647
3	33.6352	47	31	16	0.94879615	0.9375
4	39.9784	44	33	11	0.97334565	1
5	39.2166	39	28	11	0.94386898	1
6	39.234	36	22	14	0.72915741	0.92857143
7	39.2325	35	26	9	0.9708795	1
8	34.4718	35	26	9	0.98288621	1
9	45.2859	34	24	10	0.66605825	1
10	36.1061	34	19	15	0.96017181	1
11	33.635	34	28	6	0.97472685	1
12	34.4709	33	22	11	0.95717869	0.90909091
13	38.4591	30	21	9	0.90922186	1
14	38.0736	29	21	8	0.83039173	1
15	37.6877	28	19	9	0.96858159	1
16	35.7863	27	18	9	0.00203273	0
17	41.4386	25	21	4	0.94460425	1
18	40.7177	25	18	7	0.90523224	0.85714286
19	38.8537	25	21	4	0.87635221	1
20	38.8594	24	17	7	0.70115242	1
21	33.634	24	17	7	0.91623417	0.85714286
22	45.2814	23	12	11	0.92600233	1
23	42.559	23	13	10	0.86805796	1
24	40.7118	23	14	9	0.97385876	1
25	43.9459	22	16	6	0.93511035	1
26	40.7129	21	12	9	0.97270883	1
27	40.3904	21	14	7	0.59183852	1
28	35.2957	20	12	8	0.95844562	1
29	35.2945	20	14	6	0.95077553	1
30	43.2611	18	10	8	0.95160621	0.875
31	42.5598	18	12	6	0.72236229	1
32	42.5145	18	13	5	0.0786753	0
33	37.2802	18	12	6	0.74090959	1
34	45.2909	17	12	5	0.63115101	1
35	42.5644	17	9	8	0.80766985	0.875
36	36.1079	17	15	2	0.98797434	1
37	35.7829	17	11	6	0.00975675	0
38	34.1271	17	15	2	0.00016447	0
39	44.6246	16	11	5	0.706916	1
40	38.4755	16	5	11	0.92616534	1
41	38.4608	16	12	4	0.98579616	1
42	37.6866	16	8	8	0.99781483	1
43	44.6239	15	10	5	0.65576554	1
44	38.4752	15	10	5	0.91771861	1

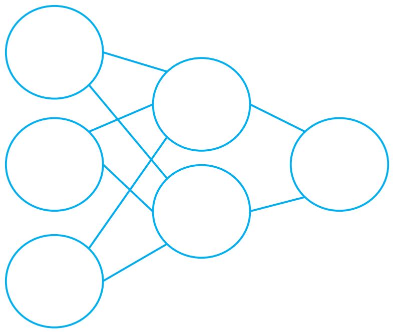
Tips:

- Visualize each input feature vs target value.
- Cluster each input feature and compare to target value

SOLUTION ARCHITECTURE



CUSTOM CNN ARCHITECTURE



	Low	High
Conv Layers	2	4
Conv size	16	256
FC Layers	1	3
FC size	16	512
Dropout	0.2	0.3
Filter size	2x2	3x3
Pooling	Max	Avg
Conv style	conv-conv-pool	conv-conv-conv-pool

Different inits, such as RandomNormal, lecun_normal, glorot_norm, he_uniform, etc.

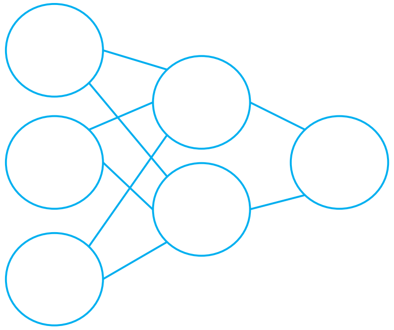
image preprocessing
xder = np.array([[-1,0,1],[-2,0,2],[-1,0,1]])
yder = np.array([[1,2,1],[0,0,0],[-1,-2,-1]])
smooth = np.array([[1,1,1],[1,5,1],[1,1,1]])
xder2 = np.array([[-1,2,-1],[-3,6,-3],[-1,2,-1]])
yder2 = np.array([[-1,-3,-1],[2,6,2],[-1,-3,-1]])
gaussian filter, binary filter, LEE filter

Algo:

- Random search to find top 10 architectures based on 4-fold CV
- Loop through each of top 10 architectures and apply image filters

Results: Best single model: 0.18, ensemble: 0.14

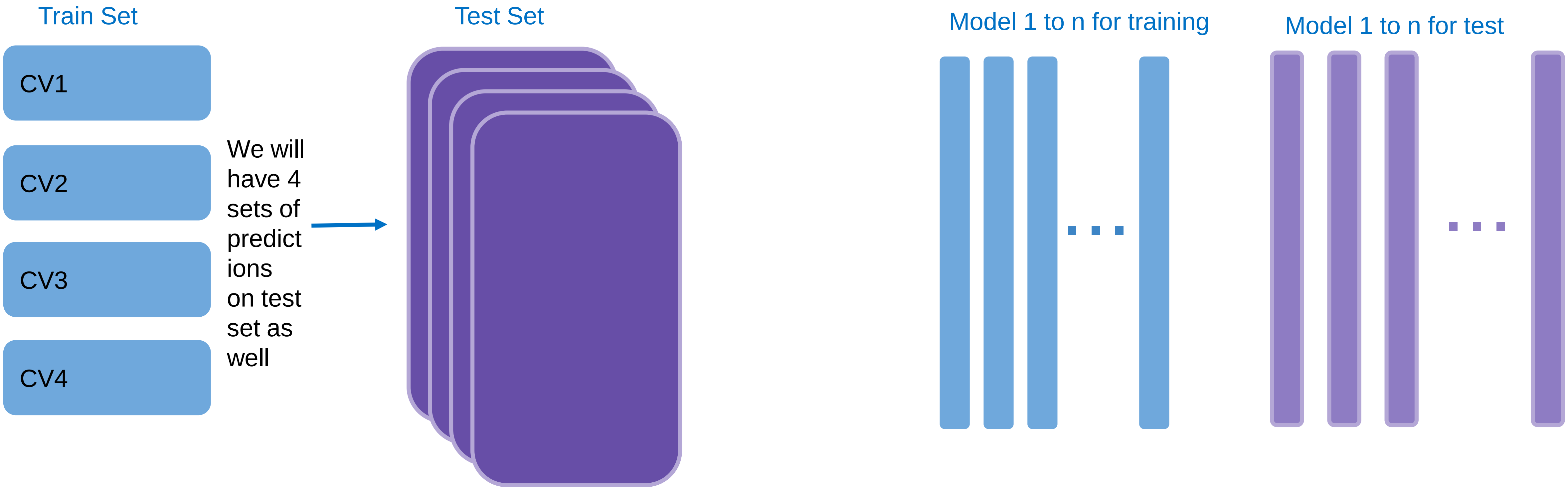
VGG16 ARCHITECTURE



	Low	High
FC layers	2	3
FC size	256	1024
Pooling	GlobalMax	Max

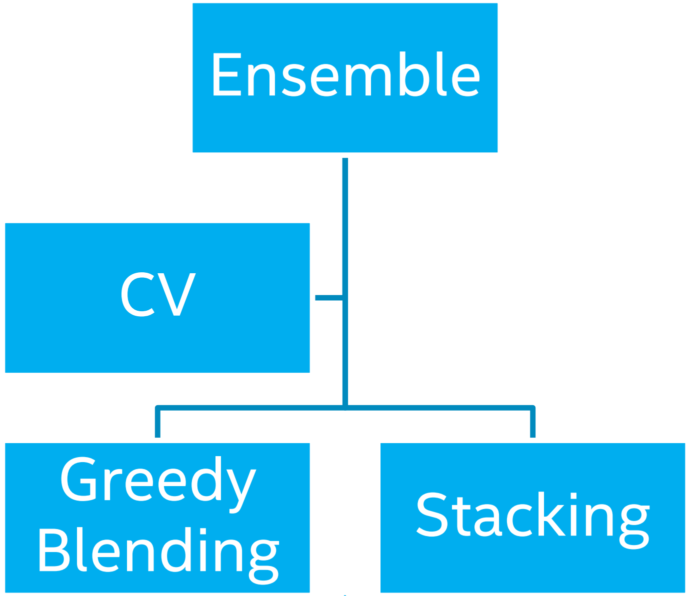
- VGG image preprocessing - synthesize third channel with random weighting

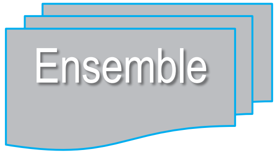
STACKING OVERVIEW

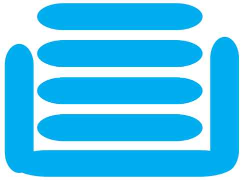


Stacking uses hold-out predictions from a model as features for a different model

ENSEMBLING





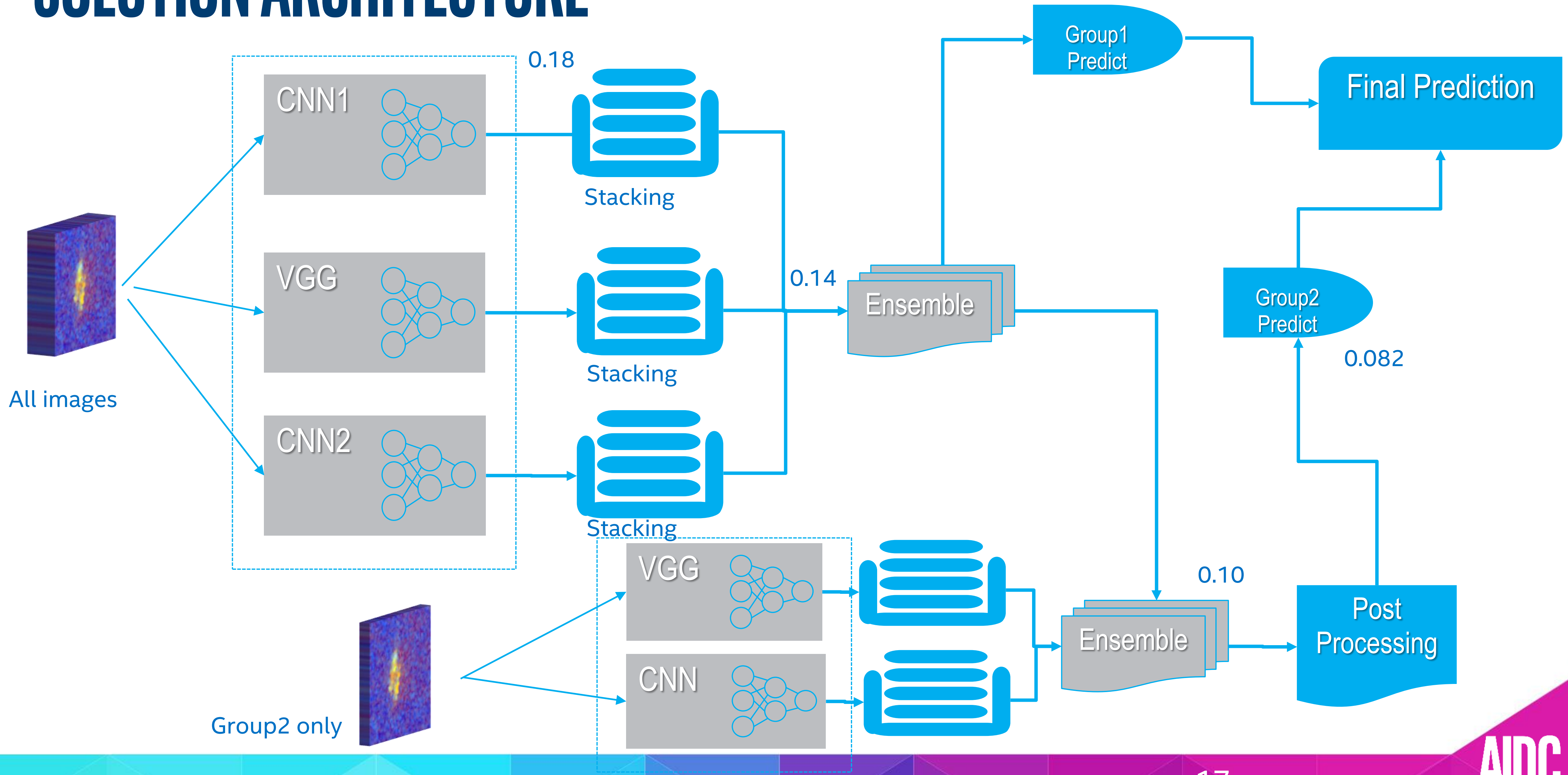


- Sort each prediction according to CV score
- Scan from best to worst, add prediction in only if median cv score improves
- Return median of all added predictions

- Using CV predictions from CNN, VGG, and Xgboost of image stats as L1 features
- Create new 2 layer CNN as level 2 stacker for final prediction

Final Prediction

SOLUTION ARCHITECTURE



POST PROCESSING

- For incorporating learning that base models don't handle
  - Group identical inc_angles
  - Unsupervised inc_angle clustering
  - inc_angle trending
  - log_loss clipping

SOLUTION DIFFERENTIATION

1. Identified inc_angle signal
2. Very large ensemble of weak base learners
3. Separating group1 and group2 into different models
4. Careful management of log_loss

KEY LEARNINGS

<u>ICEBERG COMPETITION</u>

- EDA was key
- Diversity and ensembling
- Great to team up, especially cross geo

<u>COMPETITIONS IN GENERAL</u>

- Difference from real world
- Great for learning
- Outstanding for collaboration

intel® AI