Artificial Intelligence Certification in Operational Environments

Oct. 29, 2020

Tyler Cody*, Stephen Adams, Peter Beling
University of Virginia

Erin Lanus*, Adam Edwards, Laura Freeman
Virginia Tech

Sayyed Ahamed, **Sachin Shetty***
Old Dominion University

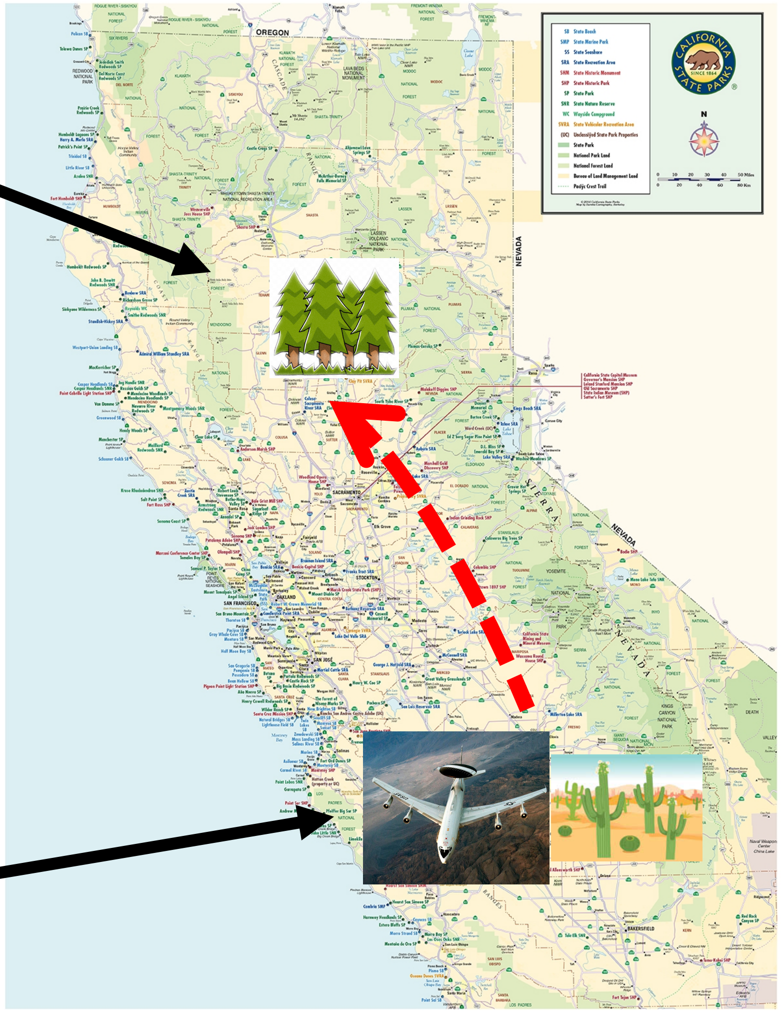
Speakers*

Setting

Mission to detect in Northern California

- Can we anticipate/detect a drop in performance?
- What can we do about it?

Trained to detect planes in Southern California

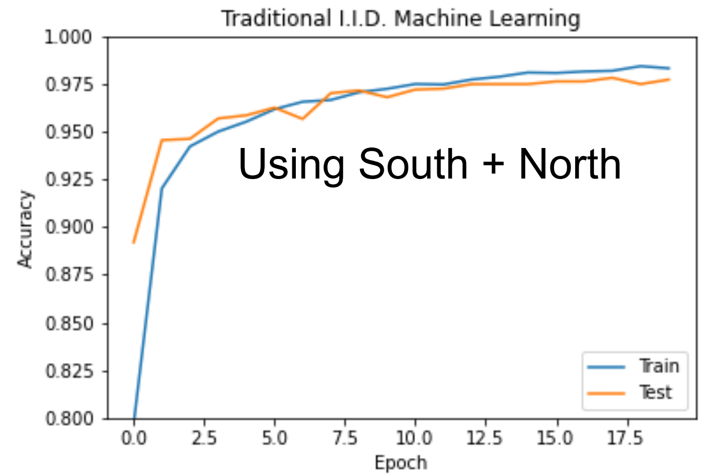


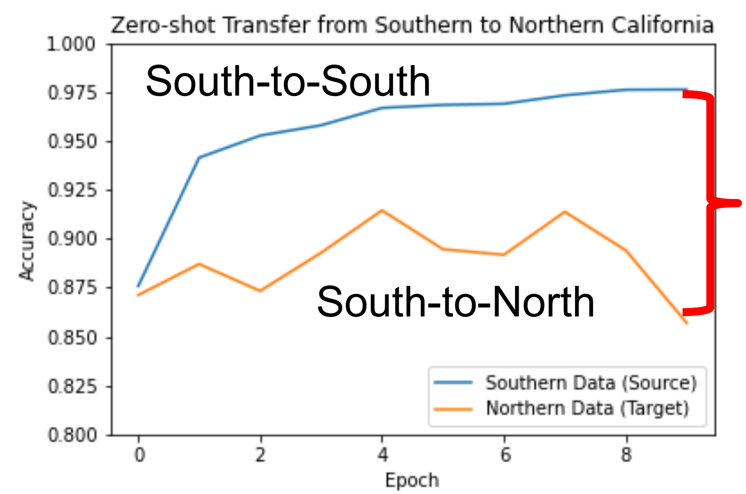
Setting – Transfer Learning

‘Planes-net’ Case Study

- Learn to detect aircraft in Southern California (*source*) ~ 22k images
- Transfer model to Northern California (*target*) ~ 10k images
- Performance drops significantly

- Can we anticipate/detect a drop in performance?
- What can we do about it?

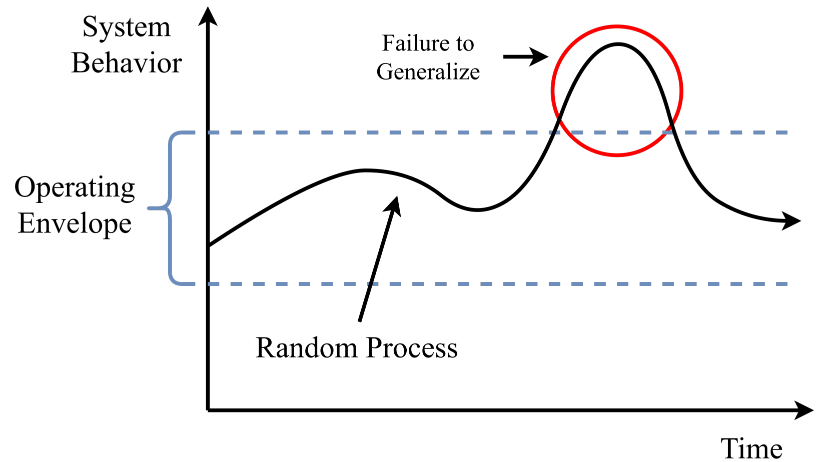




Approach

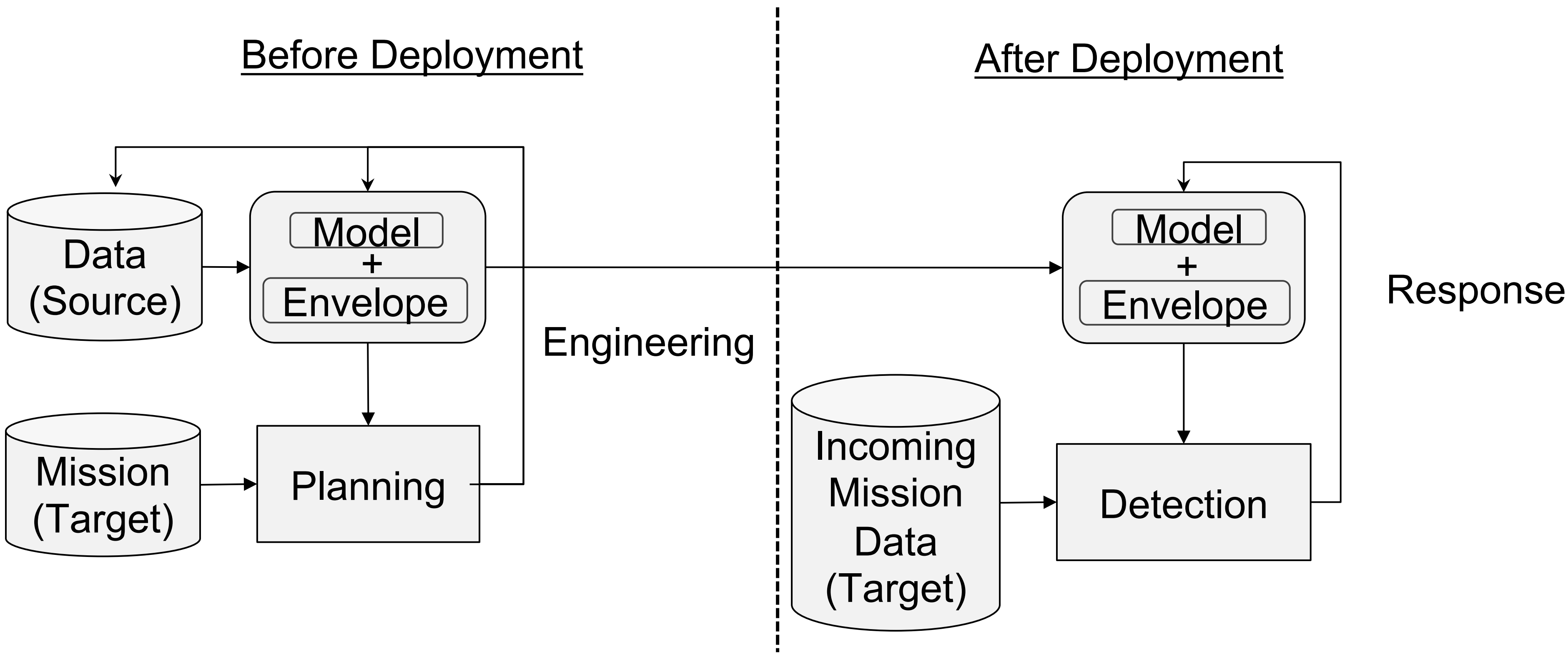
- Can we anticipate/detect a drop in performance?
  - Operating envelopes
    - using…
    - …<u>transfer distance</u>
    - …<u>combinatorial coverage</u>
- What can we do about it?
  - Collect data, search model zoo
  - <u>Transfer learning</u>

Operating Envelopes
and Time-Dependent Systems



As a system evolves over time, it risks leaving its operating envelope. A learning system leaves its envelope when **it fails to generalize** to current conditions.

Mission Scenario



Data Terminology

Labeled := images + ground truth

Unlabeled := images

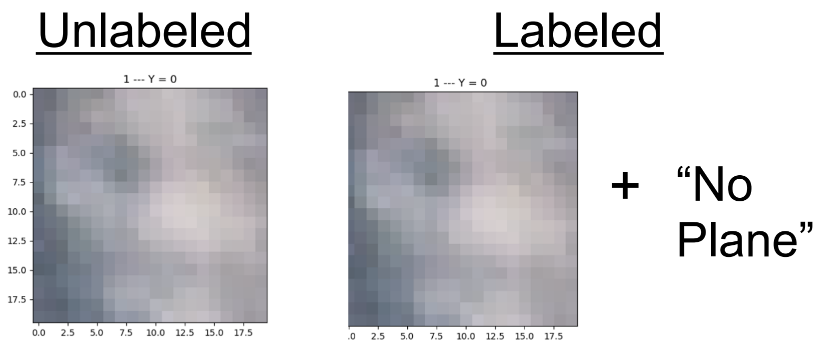
Meta-data := data related to images

Target := learning problem of interest

- *Northern California Detection*

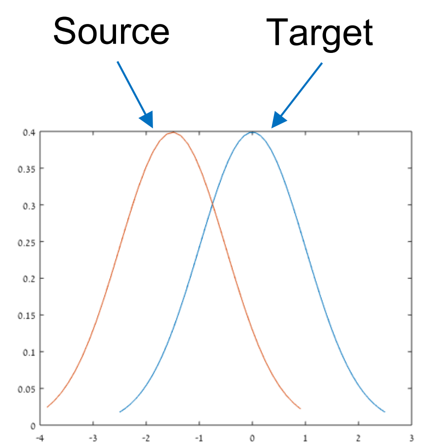
Source := related learning problem

- *Southern California Detection*



Meta-data

"Luminance", "Hue"
"Longitude, Latitude"



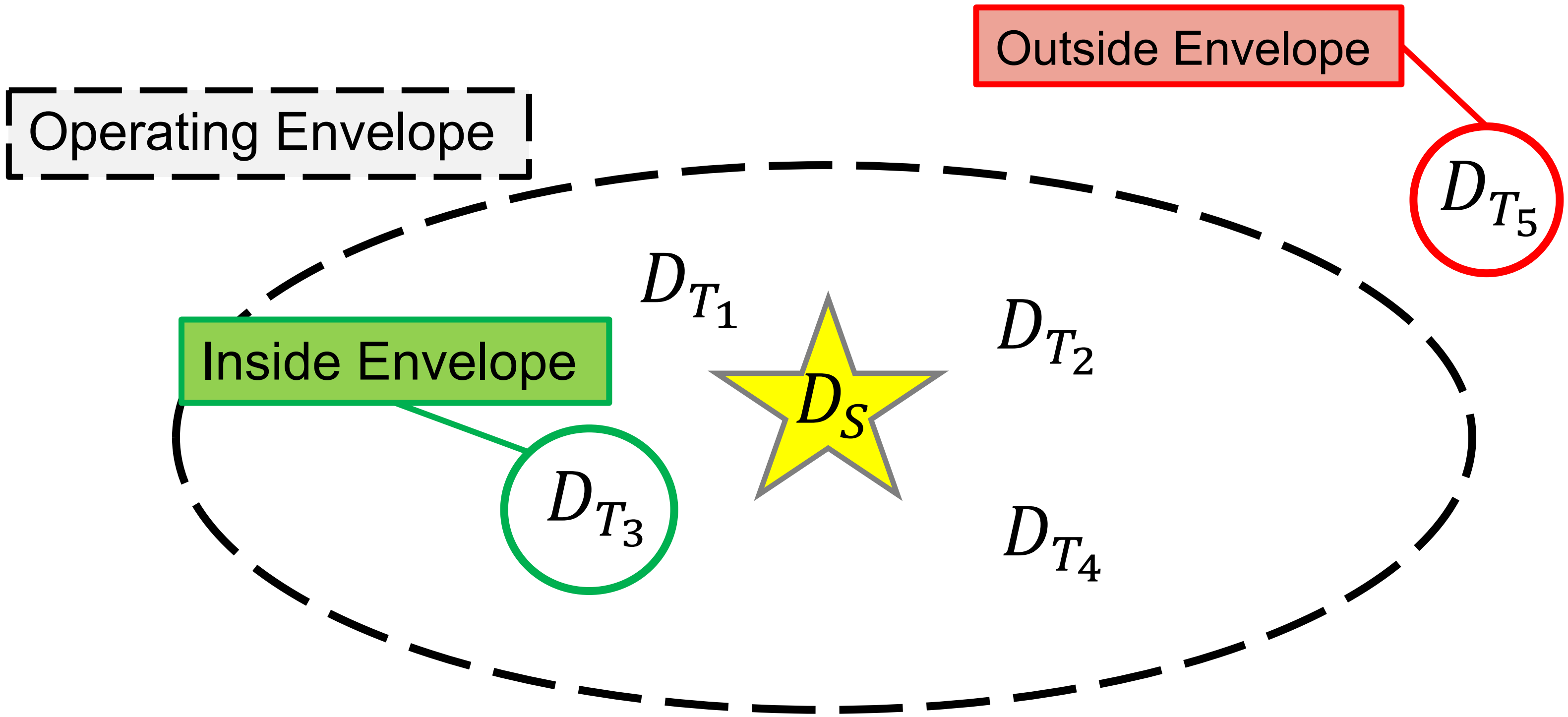
Transfer Learning

Sharing knowledge from *source* to help learn in *target*, where source and target are **distributed differently**

Operating Envelopes

Operating envelope := is the set of all systems to which we can generalize

In ML, 'systems' can be abstracted to data D (from the learning task and meta-data related to it).



Envelopes and Transfer Distance

From theory,

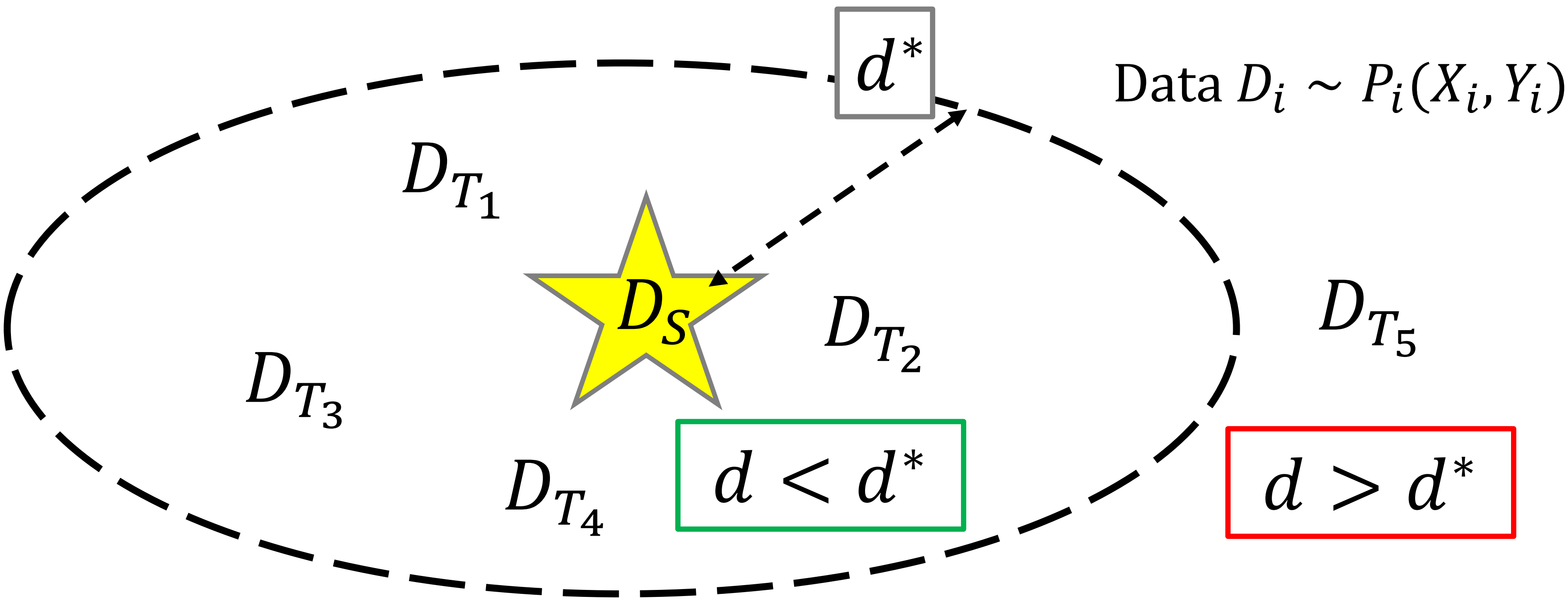
$$\epsilon_T \leq \epsilon_S + d_T + C$$

ϵ_T, ϵ_S - target and source error

d_T - transfer distance from source to target

C – constant term (VC-dimension, complexities, etc.)

d_T is **fundamental** to transferability, to determining upper-bound on error in new environments



Learning Theoretic Bounds

Ben-David, Shai, et al. "Analysis of representations for domain adaptation." *Advances in neural information processing systems*. 2007.

Theorem 1 *Let $\mathcal{R}$ be a fixed representation function from $\mathcal{X}$ to $\mathcal{Z}$ and $\mathcal{H}$ be a hypothesis space of VC-dimension d. If a random labeled sample of size m is generated by applying $\mathcal{R}$ to a $\mathcal{D}_S$-i.i.d. sample labeled according to f, then with probability at least $1-\delta$, for every $h \in \mathcal{H}$:*

$$\epsilon_T(h) \leq \hat{\epsilon}_S(h) + \sqrt{\frac{4}{m}\left(d \log \frac{2em}{d} + \log \frac{4}{\delta}\right)} + d_{\mathcal{H}}(\tilde{\mathcal{D}}_S, \tilde{\mathcal{D}}_T) + \lambda$$

Error in source ϵ_T

Constant C

Transfer distance d_T

Turn bounds into principles
$\epsilon_t \leq \epsilon_S + d_T + C$

Build empirical framework from principles

Theory-first approach!

Estimating d

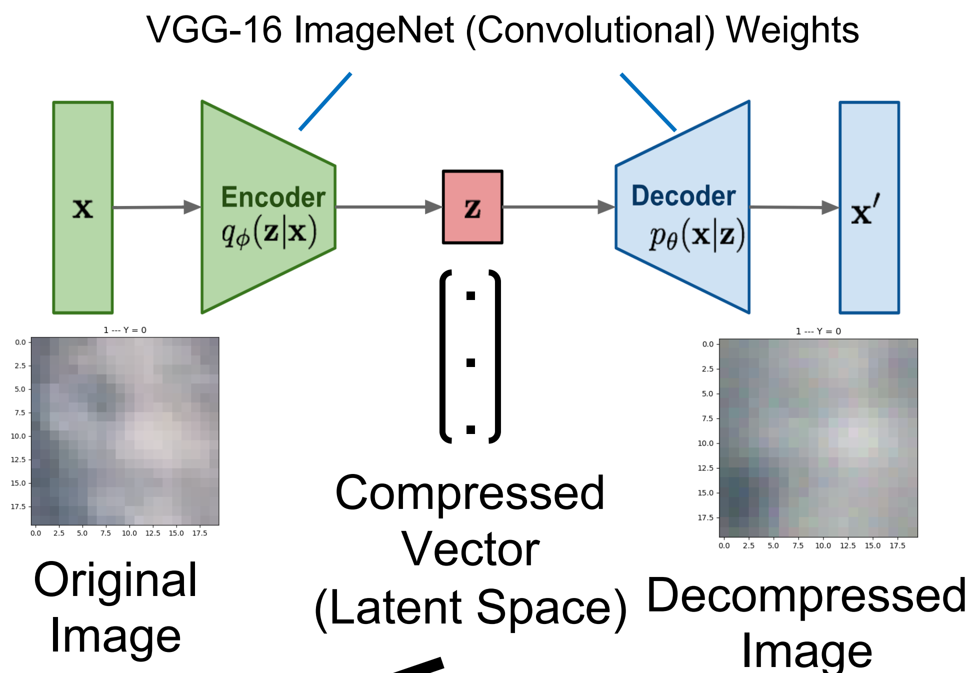
Goal: Calculate transfer distance

Problem

- Distance metrics over images are difficult to interpret/explain

Solution

1. Use latent space learned by auto-encoder to represent images
2. Calculate, visualize, and analyze distances in latent space



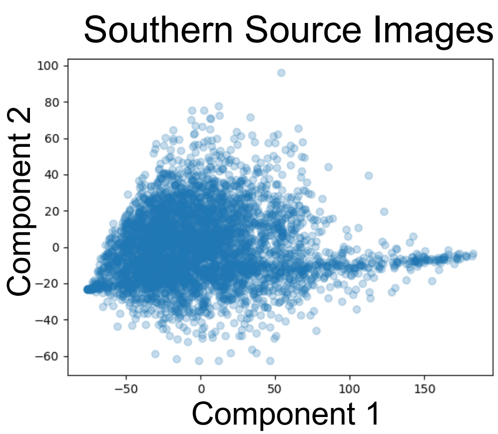
Transfer Distance Method

Principal Component Analysis → Gaussian Mixture Modeling → **Transfer Distance Calculations**

Estimating d^*

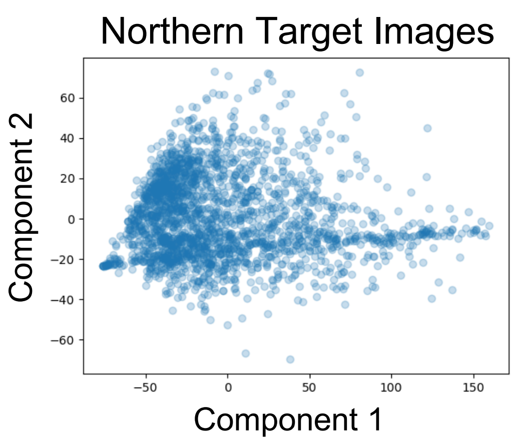
Initial estimation of d^*:

1. Split source into K-folds
2. Calculate pairwise d between folds
3. Set d^* using statistics on d between folds



Revising d^*:

Over life cycle of learning system, revise d^* using data from successful missions



Initial estimate in terms of KL divergence:

$$d^* = Normal(-11.8, 3.77)$$

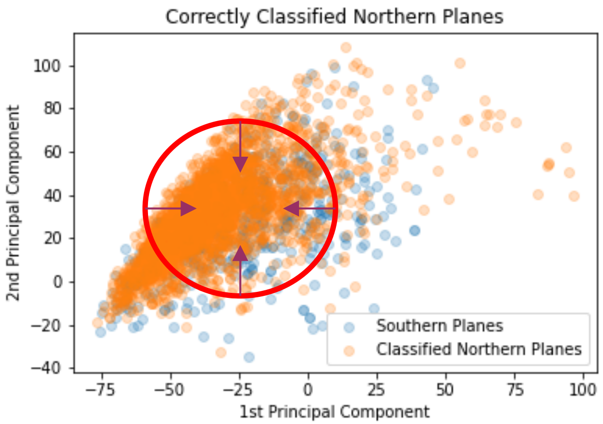
Estimate of distance to target:

$$d_T = Normal(-12.1, 4.39)$$

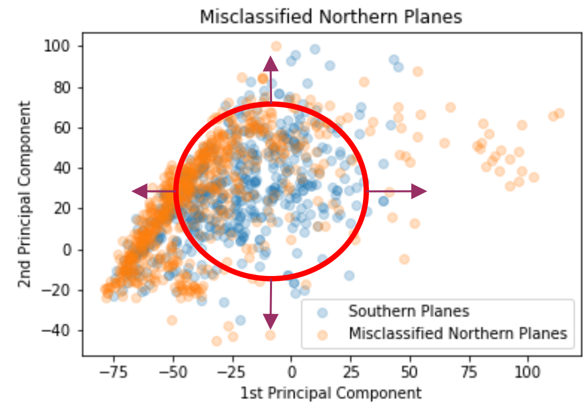
$$\boxed{d_T > d^*}$$

d_T and ϵ_T are correlated

Misclassified images have higher d_T than correctly classified images



Correctly classified Northern planes **share center-of-mass with Southern planes**



Misclassified Northern planes **do not** share center-of-mass

There are some ‘atypical’ Northern planes w.r.t. those in the South.

Table of KL d_T for true-positive, true-negative, false-positive, and false-negative cases

	$\hat{Y} = Plane$	$\hat{Y} \neq Plane$
$Y = Plane$	-8.70	$\boxed{-11.08}$
$Y \neq Plane$	$\boxed{-10.10}$	-9.97

Higher transfer distance d_T $\rightarrow$ Misclassification

d_T-based Operating Envelopes

$$N = \{D_T | d_T \leq d^*\}$$

- d_T - method for calculating transfer distance
- d^* - estimation of transfer distance threshold
- $\epsilon_T \propto d_T$ - validate target error and transfer distance are correlated

Properties:

- Theory-based
- Classifier-agnostic
- Label-free

Next, we will show how meta-data can extend this *learning theoretic* envelope.

University of Virginia
Old Dominion University
Virginia Tech

Multi-Dimensional Envelopes

Meta-data describe contexts in which images collected

Planes Metadata Columns and Raw Metadata Values

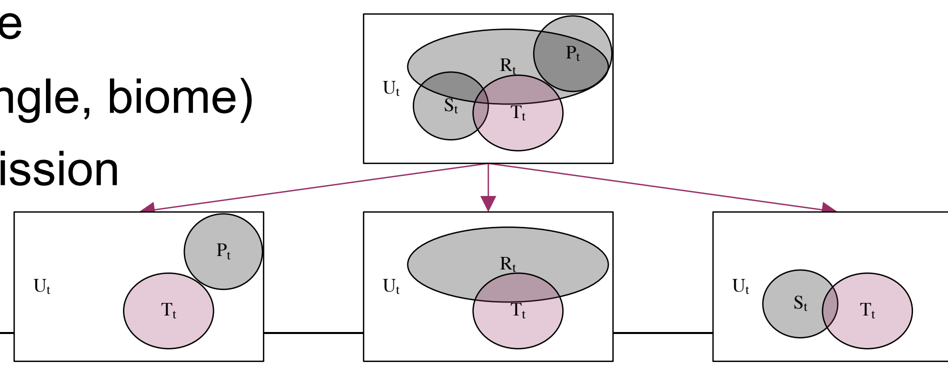
ImageID	Location	Rmean	Gmean	Bmean	Hmean	Smean	Vmean	Rvar	Gvar	Bvar	Hvar	Svar	Vvar	Class
0	Southern	0.791	0.773	0.708	0.137	0.107	0.791	0.008	0.008	0.010	0.001	0.002	0.008	Plane
1	Northern	0.833	0.834	0.787	0.183	0.064	0.839	0.013	0.012	0.013	0.004	0.001	0.012	Plane
2	Northern	0.854	0.816	0.772	0.185	0.103	0.854	0.010	0.009	0.014	0.063	0.001	0.010	Plane
3	Northern	0.679	0.693	0.670	0.279	0.035	0.693	0.004	0.004	0.004	0.003	0.000	0.004	Plane
4	Southern	0.801	0.745	0.679	0.089	0.154	0.801	0.004	0.005	0.006	0.000	0.001	0.004	Plane
5	Southern	0.884	0.833	0.762	0.097	0.142	0.884	0.011	0.011	0.014	0.002	0.002	0.011	Plane

1) Meta-data to guide calculation of transfer distance

- Use meta-data to subset images into regions, measure d_T between regions
  - By statistical effect of meta-data on trained model performance
  - Parts of target environment with/without representation in source

2) Meta-data to construct envelope when images not available

- Scenario: mission in near-future in new target environment
- No images → no transfer distance
- Known event parameters (look angle, biome)
- Use meta-data estimated from mission profile to guide model selection without collecting images



Combinatorial Interactions in Meta-data

<u>Informs Decision Making</u>
1. Guide transfer distance computation
2. Selection of model from "model zoo"
3. Targeted minimal retraining strategy

\+

<u>Explainable at 3 levels of complexity</u>
1. # or % interactions present/absent
2. Which interactions present/absent
3. Distribution of interactions

Meta-data may interact to impact performance

- *Combinatorial t-way interaction*: values assigned to *t* meta-data columns
- Computed over all $\binom{k}{t}$ combinations of columns
- Interactions present in dataset describe contexts in which model trained

Binning Scheme for Continuous Meta-data → **Planes Binned Metadata Values**

ImageID	Location	Rmean	Gmean	Bmean	Hmean	Smean	Vmean	Rvar	Gvar	Bvar	Hvar	Svar	Vvar	Class
0	0	2	2	2	0	0	2	0	0	0	0	0	0	1
1	1	2	2	2	0	0	2	0	0	0	0	0	0	1
2	1	2	2	2	0	0	2	0	0	0	0	0	0	1
3	1	2	1	1	0	0	1	0	0	0	0	0	0	1
4	0	2	2	1	0	0	2	0	0	0	0	0	0	1
5	0	2	2	2	0	0	2	0	0	0	0	0	0	1

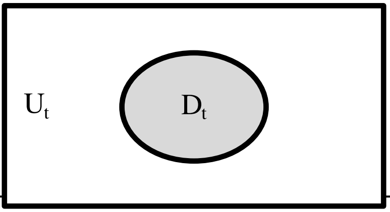
2-way combination examples:
Rmean, Gmean
Smean, Class

2-way interaction examples:
{(Rmean, 2), (Gmean, 2)}
{(Smean, 0), (Class, 1)}

Combinatorial Coverage Metric describes % of input space covered

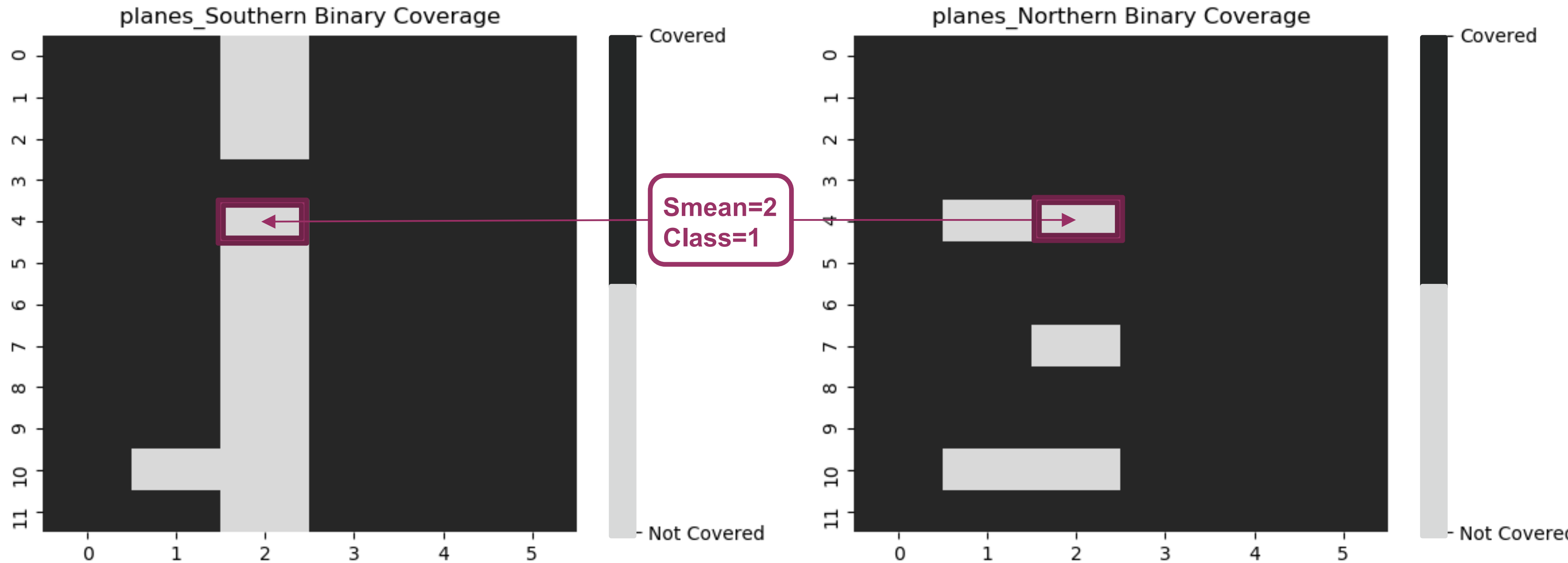
U : "universe" of meta-data
U_t : all *t*-way interactions possible
D : meta-data of training dataset
D_t : *t*-way interactions present

$$\mathrm{CCM}_t(D_t) = \frac{|D_t|}{|U_t|}$$



CCM in Planes Sets

$$Southern\ \ CCM_2 = \frac{60}{72} = .83 \qquad\qquad Northern\ CCM_2 = \frac{67}{72} = .93$$



Missing 2-way label-centric interactions:

(Rmean, 0), (Class, 1)
(Gmean, 0), (Class, 1)
(Bmean, 0), (Class, 1)
(Smean, 2), (Class, 1)
(Vmean, 0), (Class, 1)
(Rvar, 2), (Class, 1)
(Gvar, 2), (Class, 1)
(Bvar, 2), (Class, 1)
(Hvar, 2), (Class, 1)
(Svar, 1), (Class, 1)
(Svar, 2), (Class, 1)
(Vvar, 2), (Class, 1)

Missing 2-way label-centric interactions:

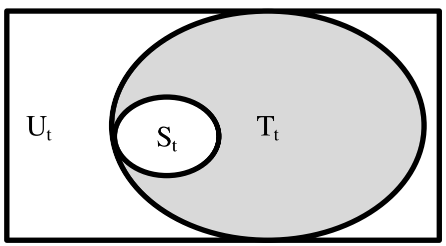
(Smean, 1), (Class, 1)
(Smean, 2), (Class, 1)
(Gvar, 2), (Class, 1)
(Svar, 1), (Class, 1)
(Svar, 2), (Class, 1)

Set Difference Combinatorial Coverage Metric

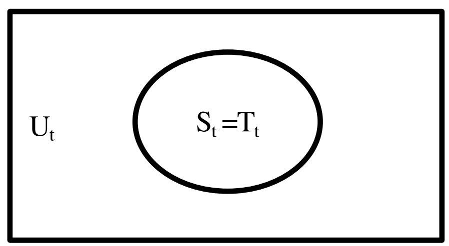
New metric measures proportion of target set interactions not covered by source set

- Describes size of difference between two sets
- Small difference → sets are more similar → expect better performance

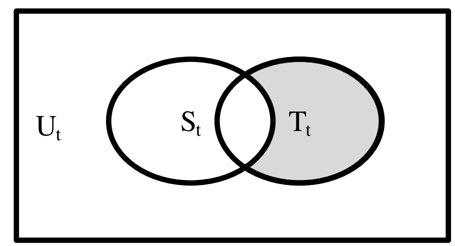
$$\mathrm{SDCCM}_t(T_t \setminus S_t) = \frac{|T_t \setminus S_t|}{|T_t|}$$



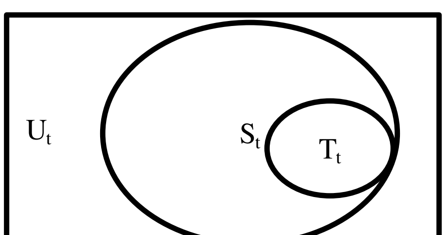
1) $0 < \mathrm{SDCCM}_t(T_t \setminus St) < 1$



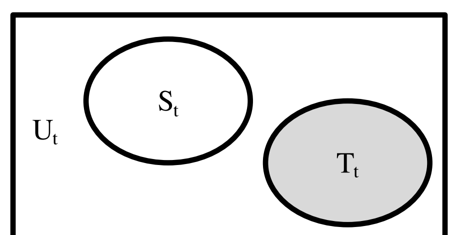
2) $\mathrm{SDCCM}_t(T_t \setminus S_t) = 0$



3) $0 < \mathrm{SDCCM}_t(T_t \setminus S_t) < 1$



4) $\mathrm{SDCCM}_t(T_t \setminus S_t) = 0$

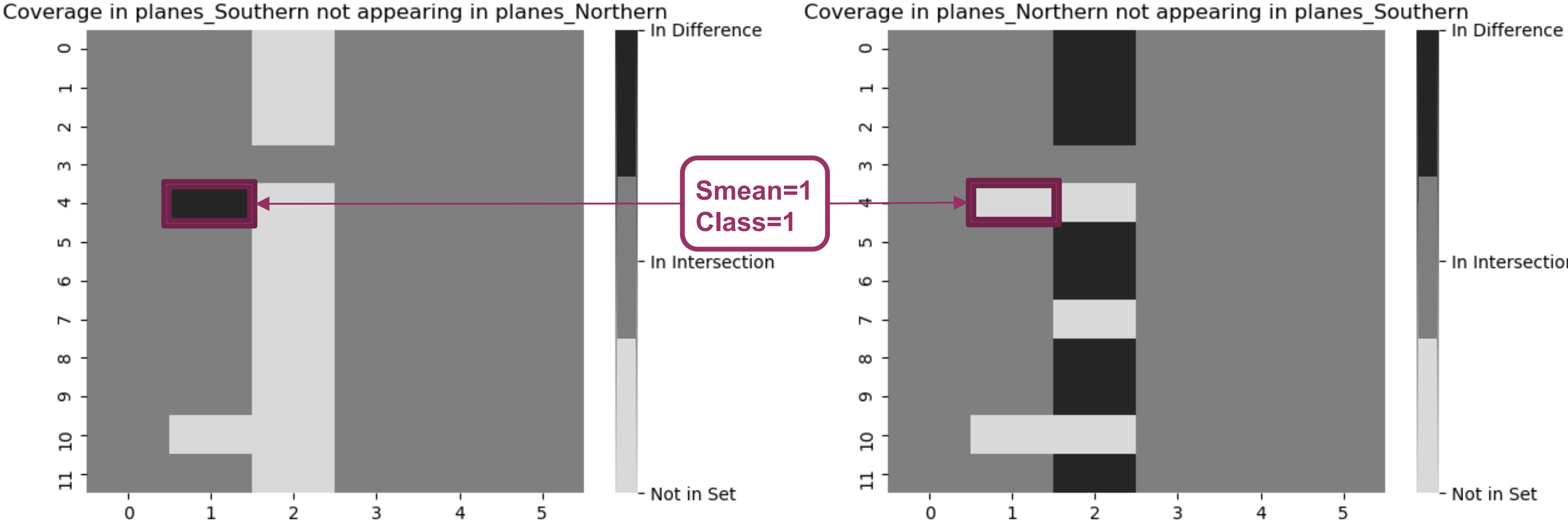


5) $\mathrm{SDCCM}_t(T_t \setminus S_t) = 1$

SDCCM in Planes Sets

$$SDCCM_2(\text{Southern}\backslash\text{Northern}) = \frac{1}{60} = .02$$

$$SDCCM_2(\text{Northern}\backslash\text{Southern}) = \frac{8}{67} = .12$$



2-way label centric interactions in set difference:

(Smean, 1), (Class, 1)

2-way label centric interactions in set difference:

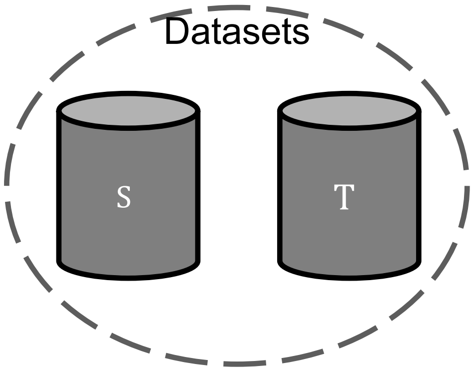
(Rmean, 0), (Class, 1)
(Gmean, 0), (Class, 1)
(Bmean, 0), (Class, 1)
(Vmean, 0), (Class, 1)
(Rvar, 2), (Class, 1)
(Bvar, 2), (Class, 1)
(Hvar, 2), (Class, 1)
(Vvar, 2), (Class, 1)

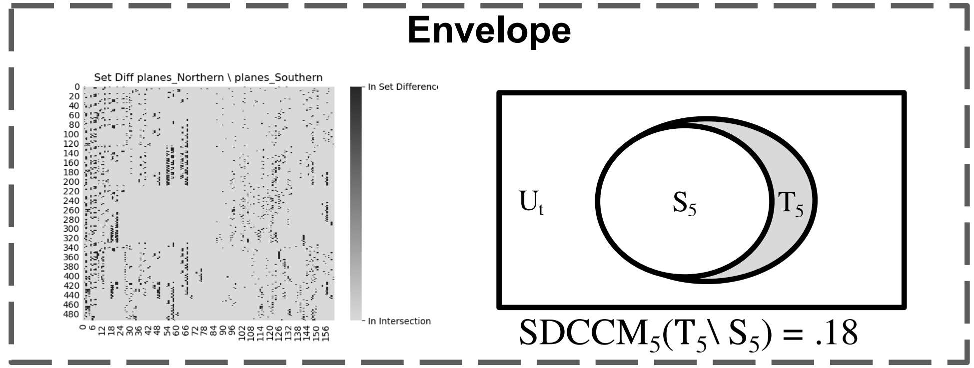
Most Southern interactions covered in Northern
Many Northern interactions not covered in Southern
→ SDCCM results correlate to presence/absence of transfer learning problem

All missing interactions have "Plane" vs. "No Plane"
→ Describe "atypical" Northern plane images in contexts not seen in South?

Describes contexts missing from model
→ Images difficult to classify
→ Informs minimal re-training strategy

SDCCM for Targeted Retraining





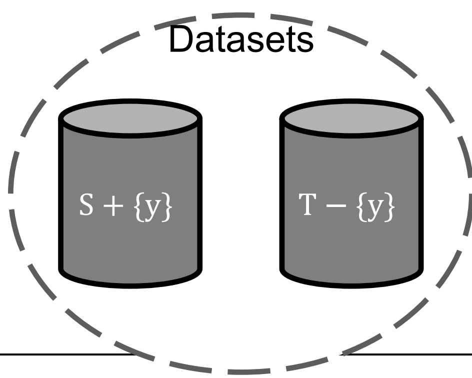
Automatically Identify Interactions $\{x : x \in T_t \setminus S_t\}$

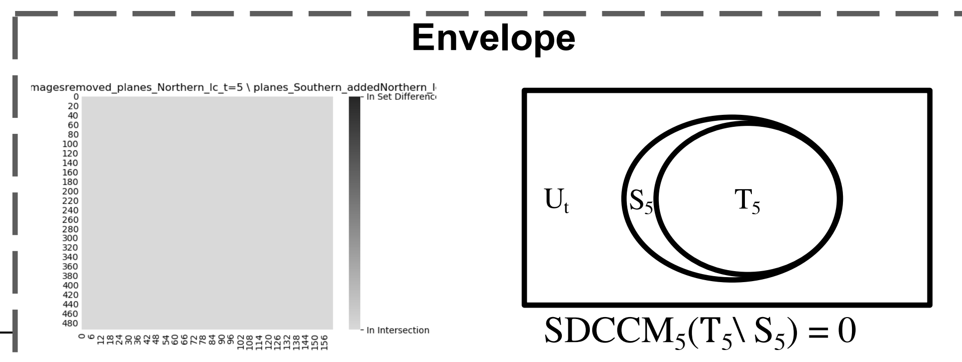
Automatically Identify Images $\{y : x \in Metadata(y)\}$

Create train/test sets

Training set $S + \{y\}$

Test set $T - \{y\}$





SDCCM, Transfer Learning and Fine-tuning - AI Certification Process

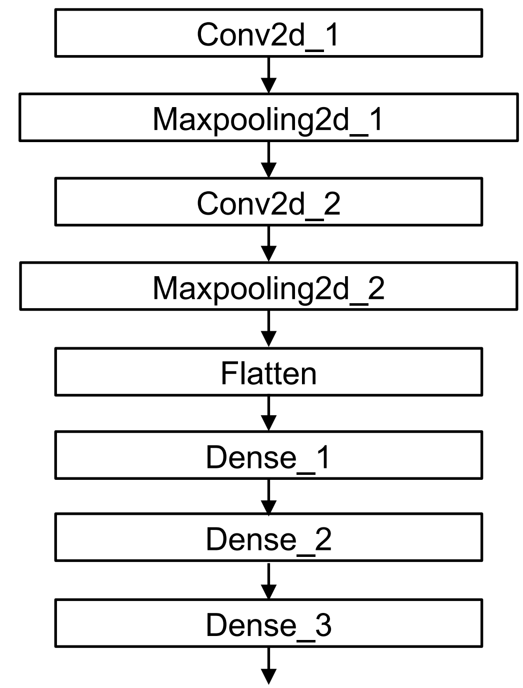
- Operator provides desired metrics for certification in target environments– Precision, Recall, F1-score and Accuracy

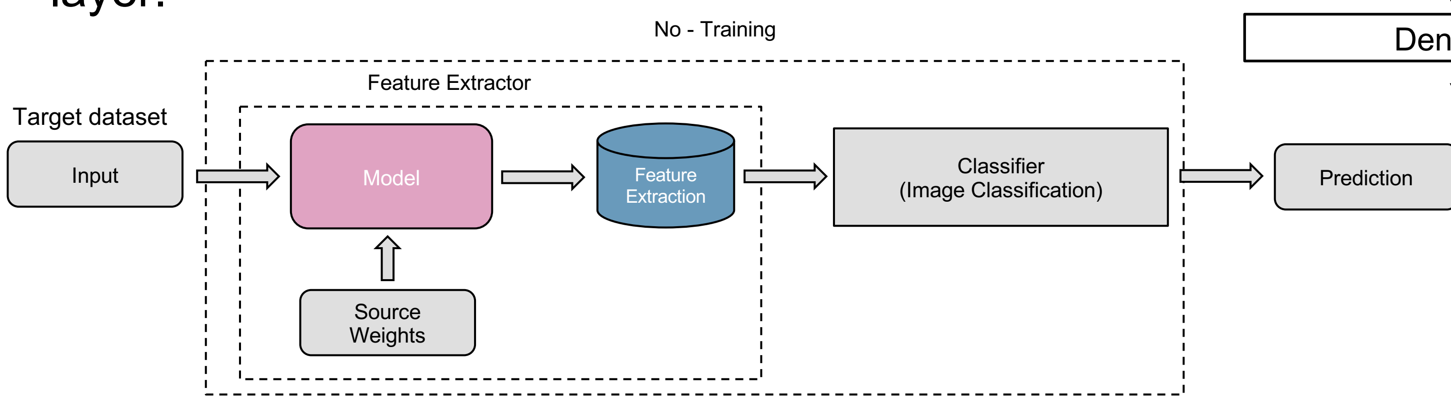
- AI Certification Process on Planes dataset
  - **Step 1** – Can **pre-trained model** trained on Southern California provide desired metrics when evaluated on Northern California?
  - **Step 2** - Can **pre-trained model** trained on Southern California **plus set difference interactions** from Northern California provide desired metrics?
  - **Step 3** – Can **fine tuning** of the **pre-trained model** trained on Southern California **plus set difference interactions** from Northern California provide desired metrics?

Transfer Learning

- Split the source dataset randomly (train-90%, validation-10%).
- Train a model on the source dataset (Southern California).
- Prediction:
  - Take target dataset (Northern California) as a test model.
  - The Source task and the target task are same, there is no need to add a classifier layer.
  - Prediction made with the same classifier layer.

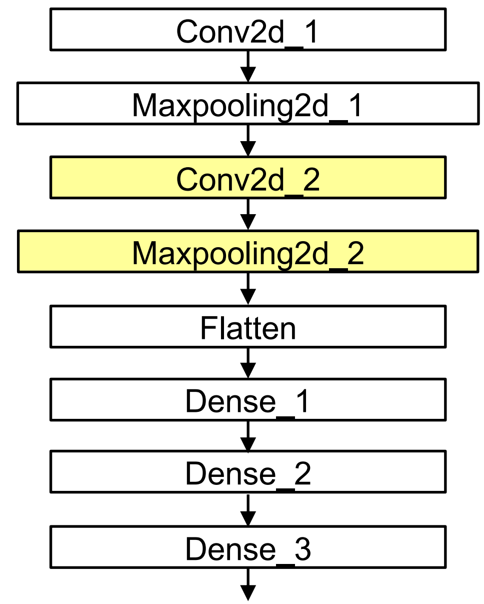
Model architecture





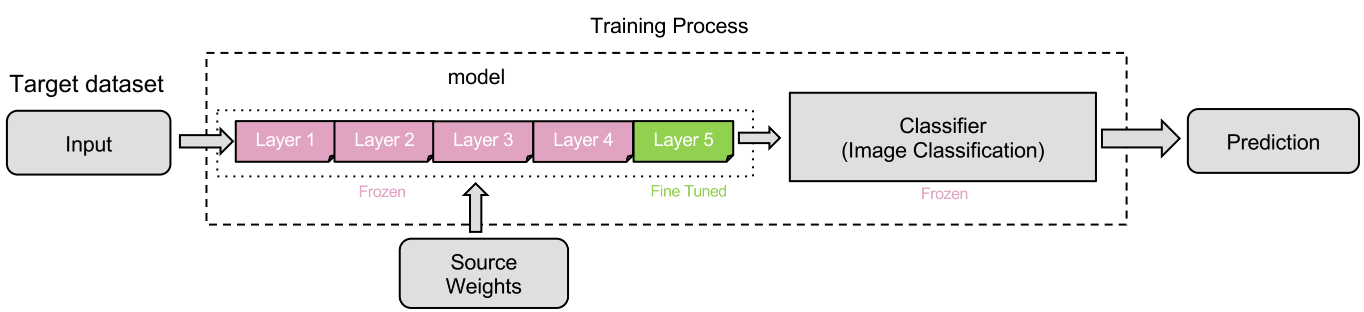
Fine Tuning

Model architecture:

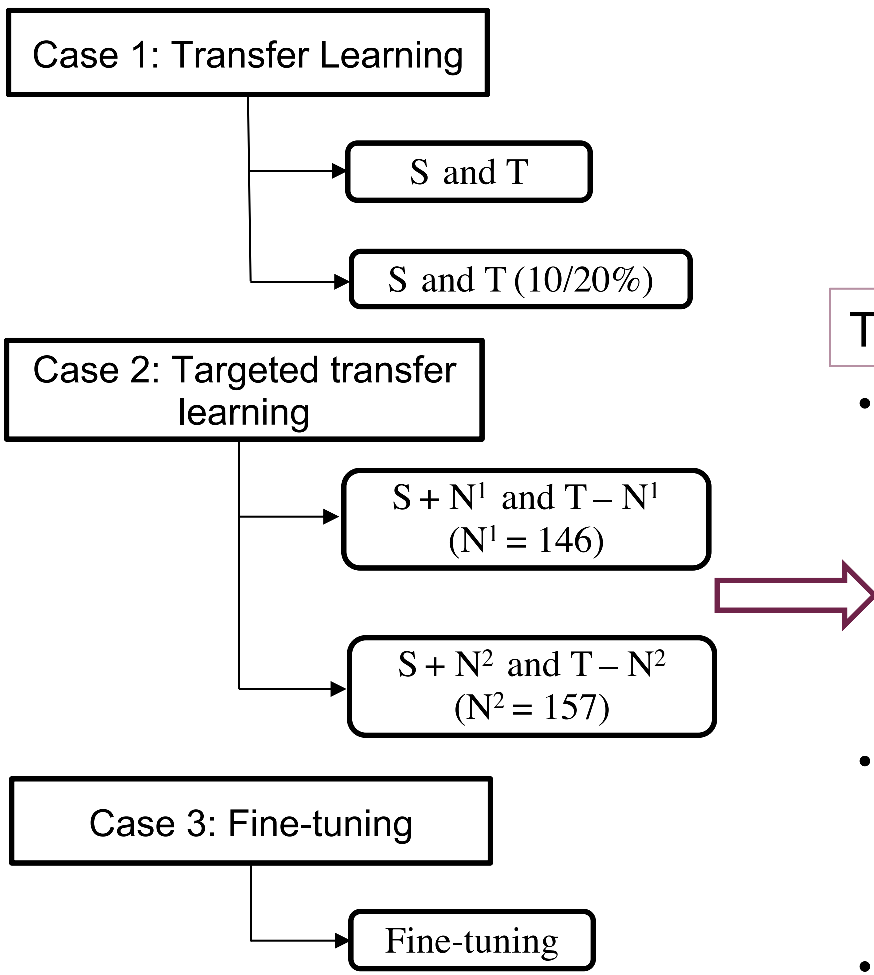


- Access to pre-trained network.
- Freeze weights except the last convolutional layer.
- Since the source task and target task is same, use the previous classifier layer.
- Train only the last convolutional layer with a low learning rate.
  - Split the target data set randomly (train-80%, validation-10%, test-10%)
  - Make the prediction.

Fine-tuning



AI Certification Process



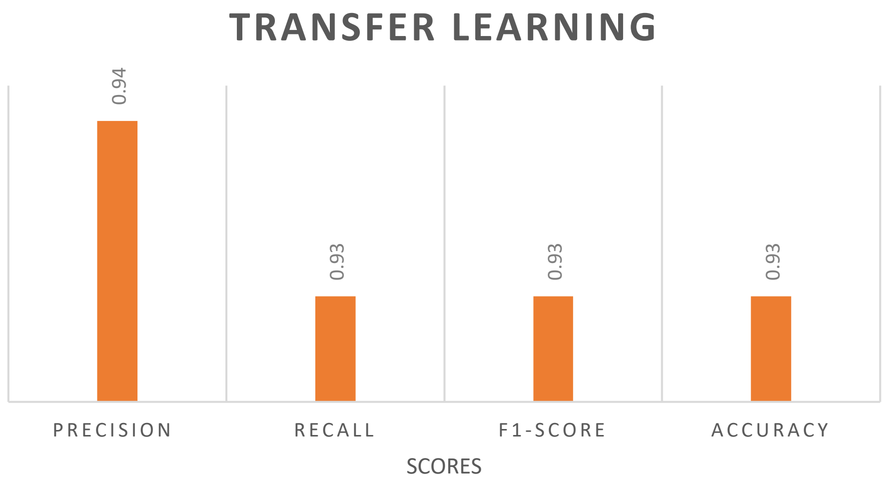
S	Source dataset [Southern California]
T	Target dataset [Northern California]

Targeted transfer leaning process

- Step 1: Identify the interactions in Northern that don't appear in Southern for t = 5 [N^1 = label centric, N^2 = not label centric]. Identify the images that contain interactions in the set difference and add them to the Southern set for training. Test on the remaining Northern set.
- Step 2: Randomly select the same number of images from the Northern set and add them to the Southern set for training. Test on the remaining Northern set.
- Step 3: Compare the performance between 1 and 2.

Case 1: Transfer Learning Results

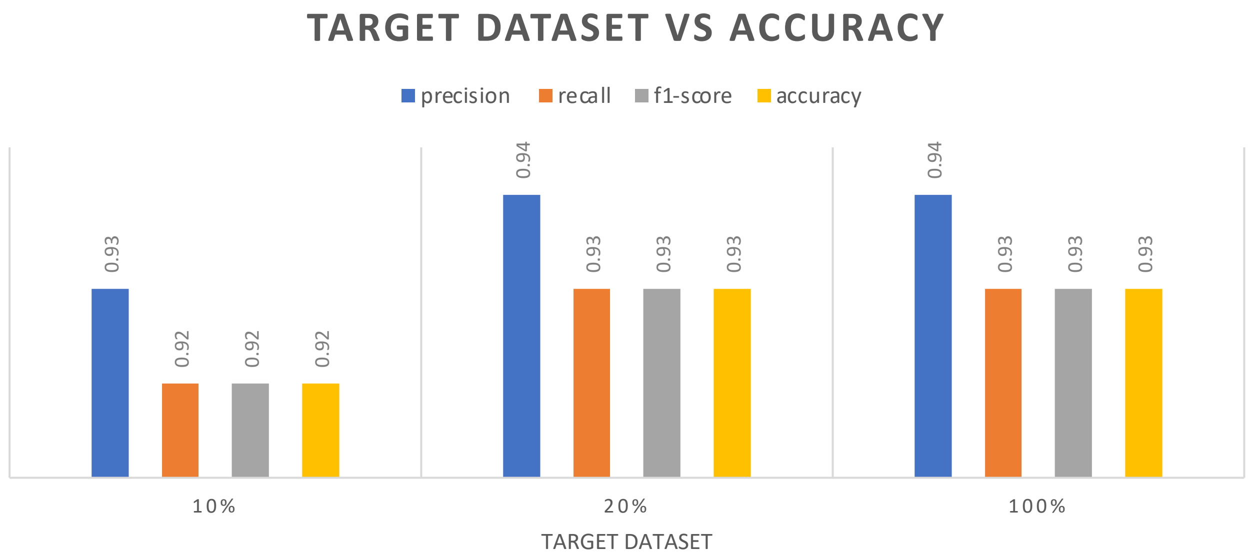
Source [Southern- train(90%), validation(10%)]
Target [Northern- test(100%)]



- Without training on the target dataset, we get around 93% accuracy using transfer learning, which is 5% less than the desired accuracy (98%).

Case 1: Transfer Learning with Varying Random Data Additions

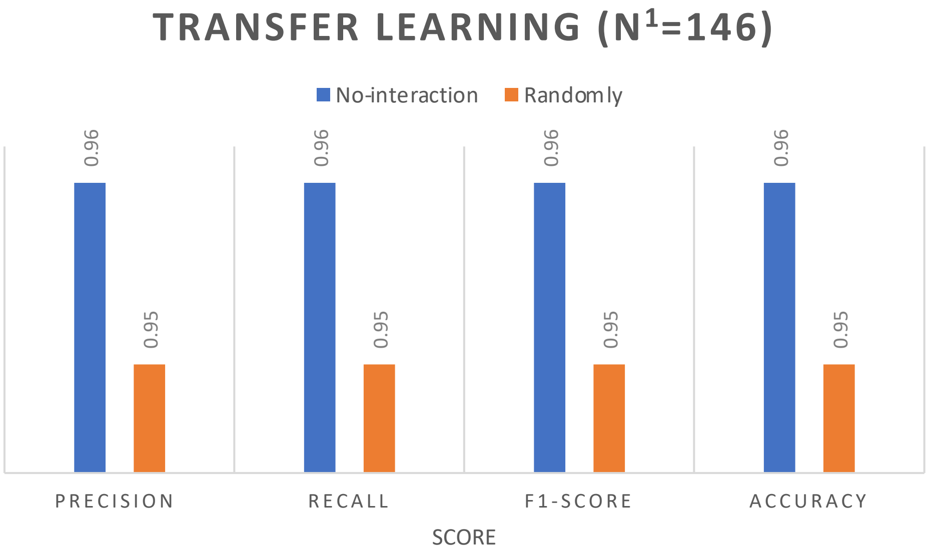
Source [Southern- train(90%), validation(10%)]
Target [Northern- test(10/20/100%)]



- Since the source and target dataset/task are the same, it is not necessary to train the model in the target space.
- Therefore, the accuracy is not very dependent on the target dataset size.

Case 2: Targeted Transfer Learning

Source [Southern- train(90% + N[1]), validation(10%)]
Target [Northern- test(100% - N[1])]

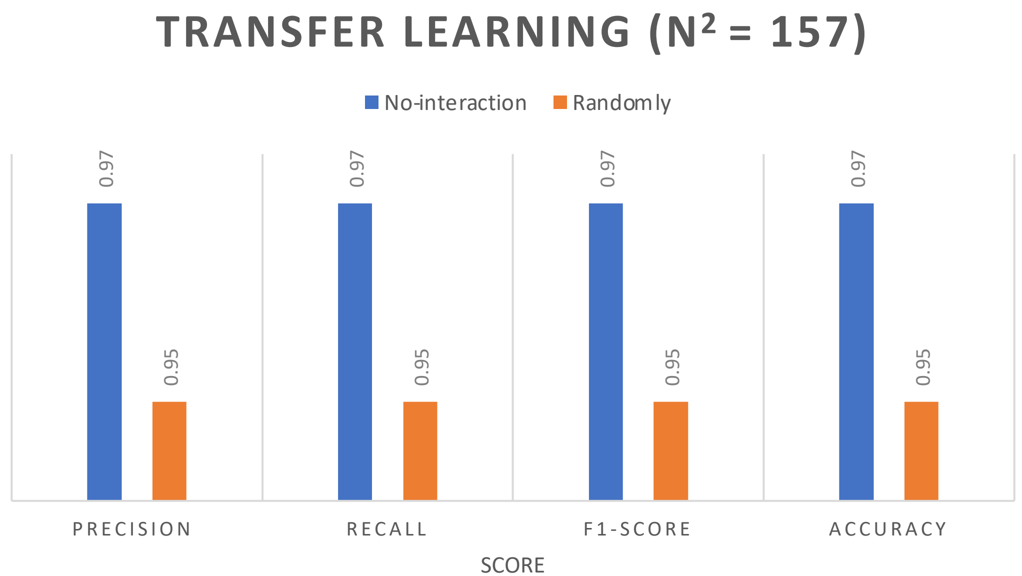


- Source (increased) – around 0.68%
- Accuracy (increased) - 2% (set difference label centric interactions)
- Accuracy (increased) - 1% (randomly)

- Information factor per image (no-interaction) – $1.36*10^{-4}$
- In set difference interaction images, around $6.85*10^{-5}$ more information factor per image

Case 2: Targeted Transfer Learning

Source [Southern- train(90% + N^2), validation(10%)]
Target [Northern- test(100% - N^2)]

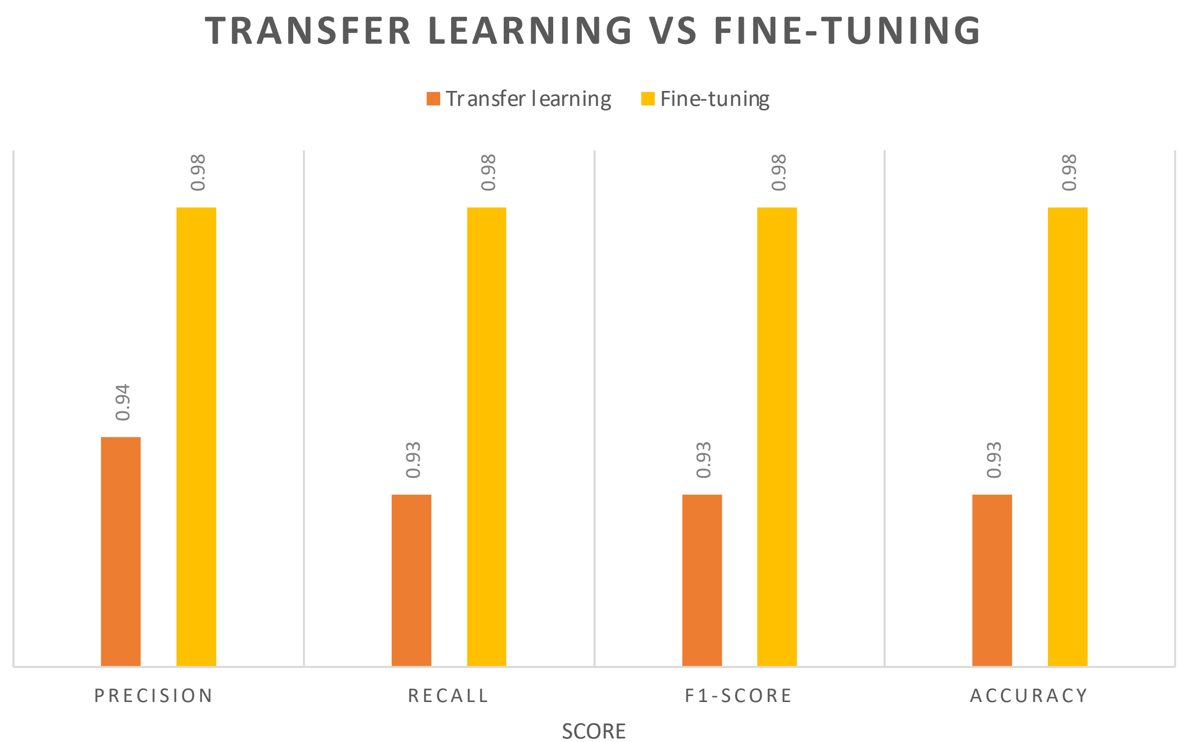


- Source (increased) – around 0.74%
- Accuracy (increased) - 3% (set difference all interactions)
- Accuracy (increased) - 1% (randomly)

- Information factor per image (no-interaction) – $1.9*10^{-4}$
- In set difference interactions images, around $1.27*10^{-4}$ more information factor per image

Case 3: Transfer Learning vs Fine-Tuning

Source [Southern- train(90%), validation(10%)]
Target [Northern- train(80%), validation(10%), test(10%)]

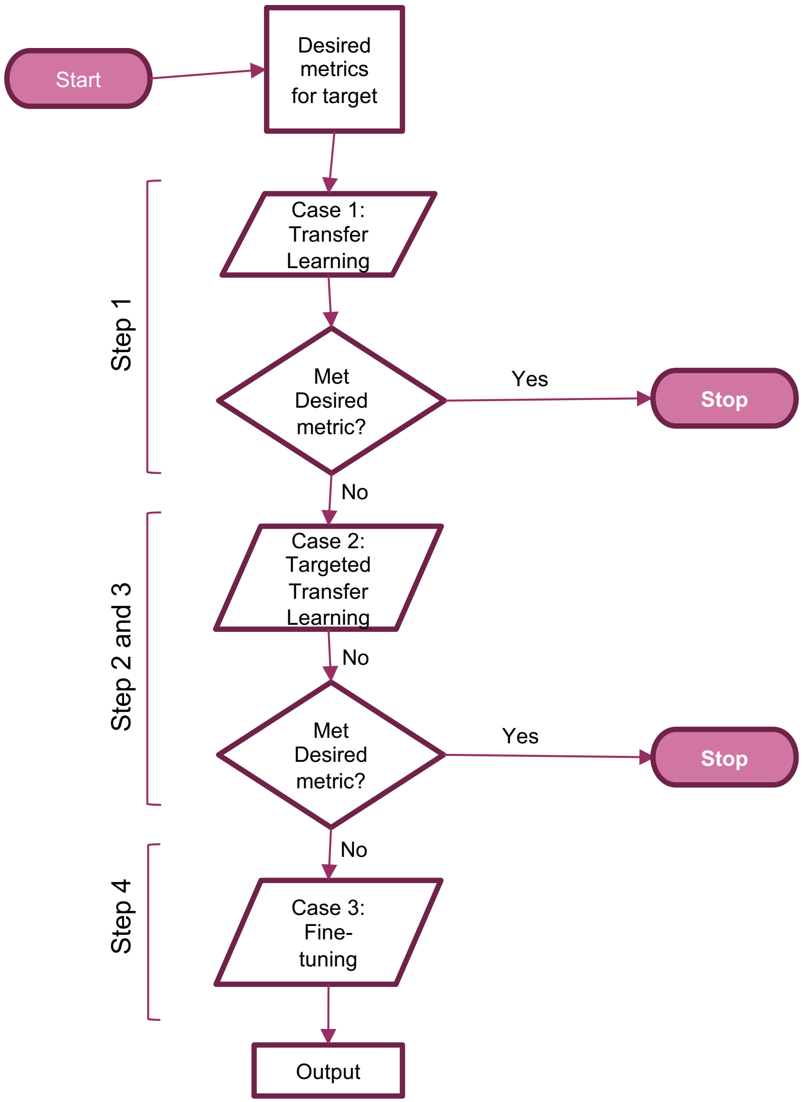


Results Summary

- If Operator's desired metrics for certification in target environments– **Precision (0.97), Recall(0.97), F1-score(0.97) and Accuracy (0.97)**

- AI Certification Process on Planes dataset
  - **Step 1** – Can **pre-trained model** trained on Southern California provide desired metrics when evaluated on Northern California?
    Precision (0.94), Recall(0.93), F1-score(0.93) and Accuracy (0.93)
  - **Step 2** - Can **pre-trained model** trained on Southern California **plus set difference interactions** from Northern California provide desired metrics?
    LC: Precision (0.96), Recall(0.96), F1-score(0.96) and Accuracy (0.96)
    All: Precision (0.97), Recall(0.97), F1-score(0.97) and Accuracy (0.97)
  - **Step 3** – Can **fine tuning** of the pre-trained model trained on Southern California **plus set difference interactions** from Northern California provide desired metrics?
    Precision (0.98), Recall(0.98), F1-score(0.98) and Accuracy (0.98)

Next Steps: Automated Transfer Learning



Goal: Automate selection of transfer learning, targeted transfer learning and fine-tuning methods to meet desired metric for target environment

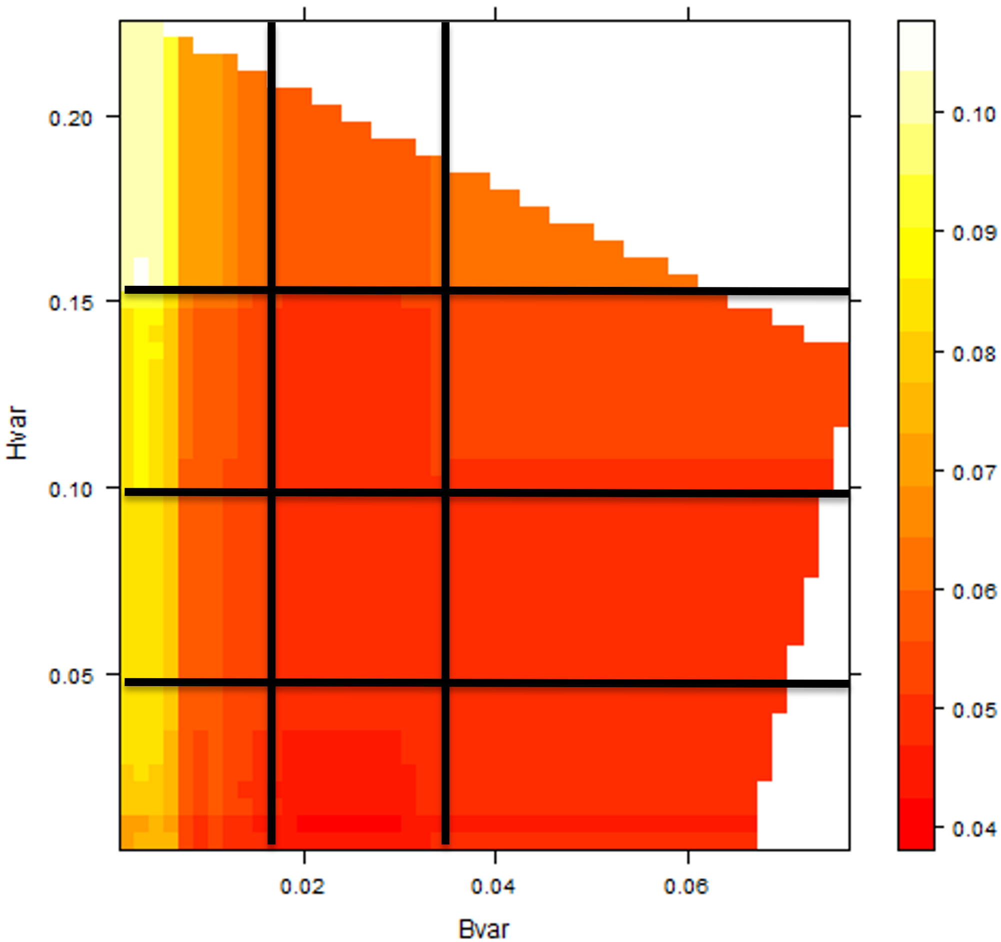
Tasks: We will develop the software that will automate the below tasks and meet the aforementioned goal

Task 1 - Identify the interactions in Target set that don't appear in source set (explore label centrism and non-label centrism/all).

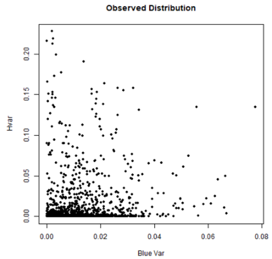
Task 2 - Identify the images that contain interactions in set difference and add them to the source for training and test on the remaining on target set.

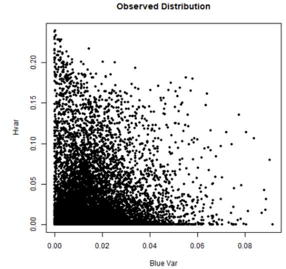
Task 3 - Develop automated fine-tuning strategy to meet desired matric for target.

Next Steps: Integrating Modelling Classifier Performance



- The two way partial dependence plot gives an indication of where we can make bins in a way that correlates to the classifier performance.





Summary: Novel Contributions

- **Decision making framework**
  - Data driven resource allocation
- **Connecting measures of transferability in latent space to metadata and model performance**
  - Allows for black-box algorithm assessment
  - Integration techniques for black-box methods
- **Application of Combinatorial Coverage to Machine Learning**
  - Metadata coverage as partial descriptor for operational envelope
  - Extension beyond single set w.r.t. universe to multi-set with directionality
- **Fine-tuning process for**
  - Achieving desired metrics for target environment
  - Satisfying computational resource usage and response requirements
- **Policy network for fine-tuning that is**
  - Model agnostic, configurable, distributed in the network

What questions will this allow us to answer?

- **Given a collection of models trained on various data sets – which models are expected to perform well in a new environment?**
  - Are there types of models that tend to perform well new environments?
  - Are there types of models that transfer well with minimal retraining?
  - What characteristics does my training data need to facilitate robust model performance in new operating environments?

- **Can we detect when a model starts operating outside of its certified operating envelope?**

- **What is the best allocation of resources to ensure certification in new environments?**
  - A model exists – just use it
  - Invest in ensembling or orchestration to combine benefits of multiple existing models
  - Invest in fine tuning a training model in the target environment
  - None of these will work – invest in additional data collection, retraining, or accept risk

Increasing Economic Impact

Deliverables

- Slides
- Code for each portion of analysis
- Summary Report

Backup Slides

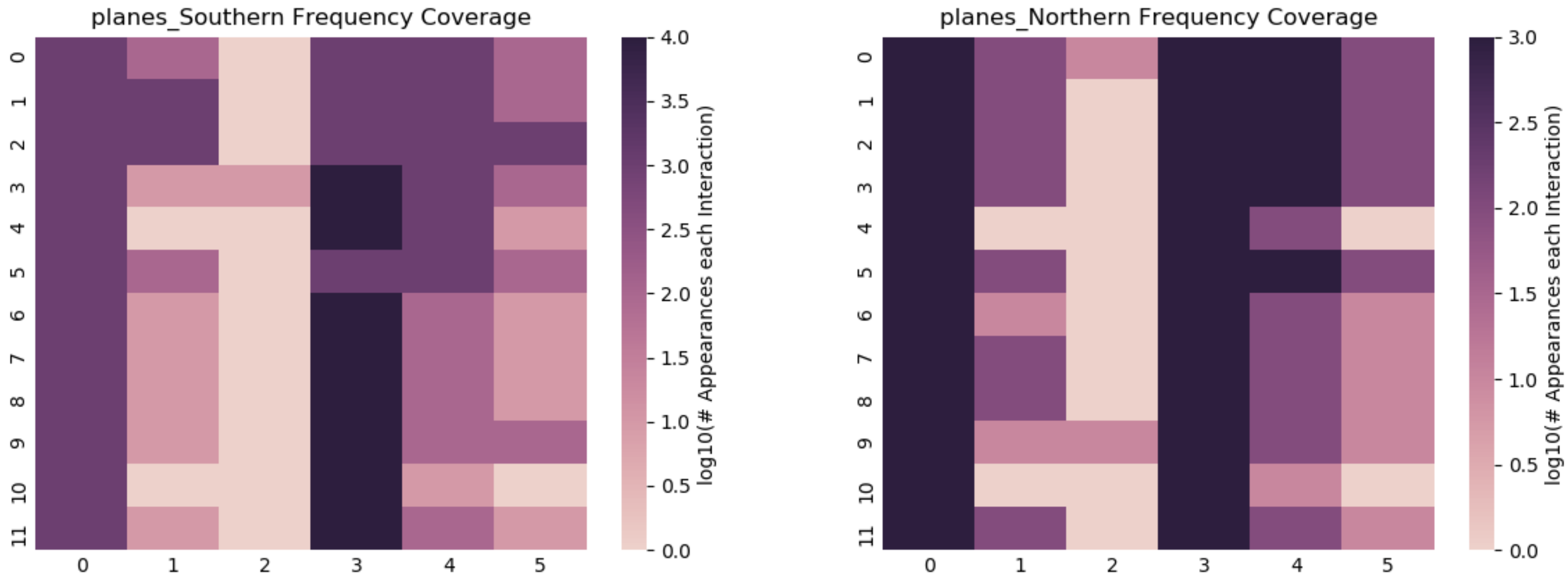
- **Future Directions**

Future Work: Difference to Support Distance

- **Set Difference Combinatorial Coverage and Transfer Distance**
  - Two metrics to describe a model's operating envelope
  - In Planes-net, **both metrics** identified that outlier region contained Northern images containing planes
- **Are they describing the same phenomena?**
  - Use SDCCM to identify images in set difference
  - Compute transfer distance on those images
  - Check for correlation
- **If yes, use SDCCM to subset images for computing transfer distance**
- **If no, might be describing different dimensions of the envelope**
  - Use them together to get a more accurate description

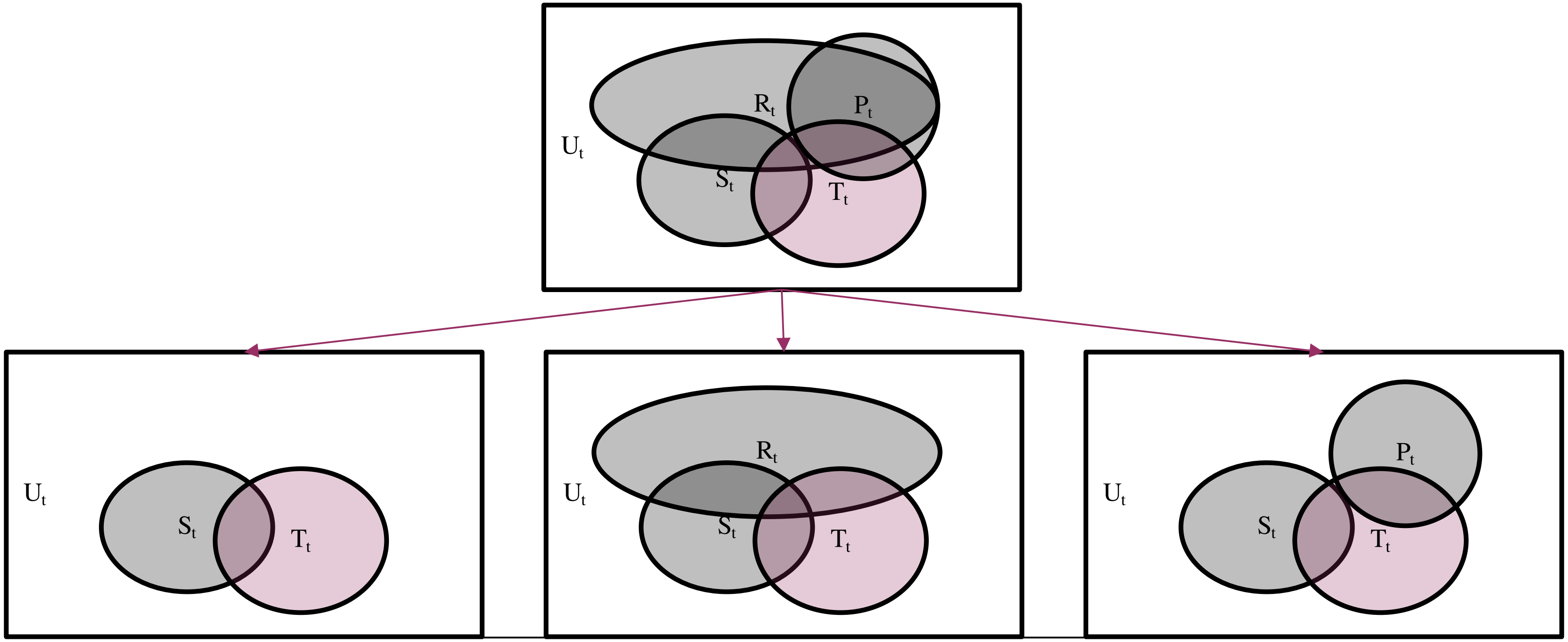
Future Work: Coverage Distribution

- **Coverage is currently a binary metric: present/absent**
- **Distribution of coverage does more to describe contexts on which model trained**
  - Current: measure and plot distribution
  - Future: develop metrics for set difference distribution/relative frequency, evaluate use in transfer learning

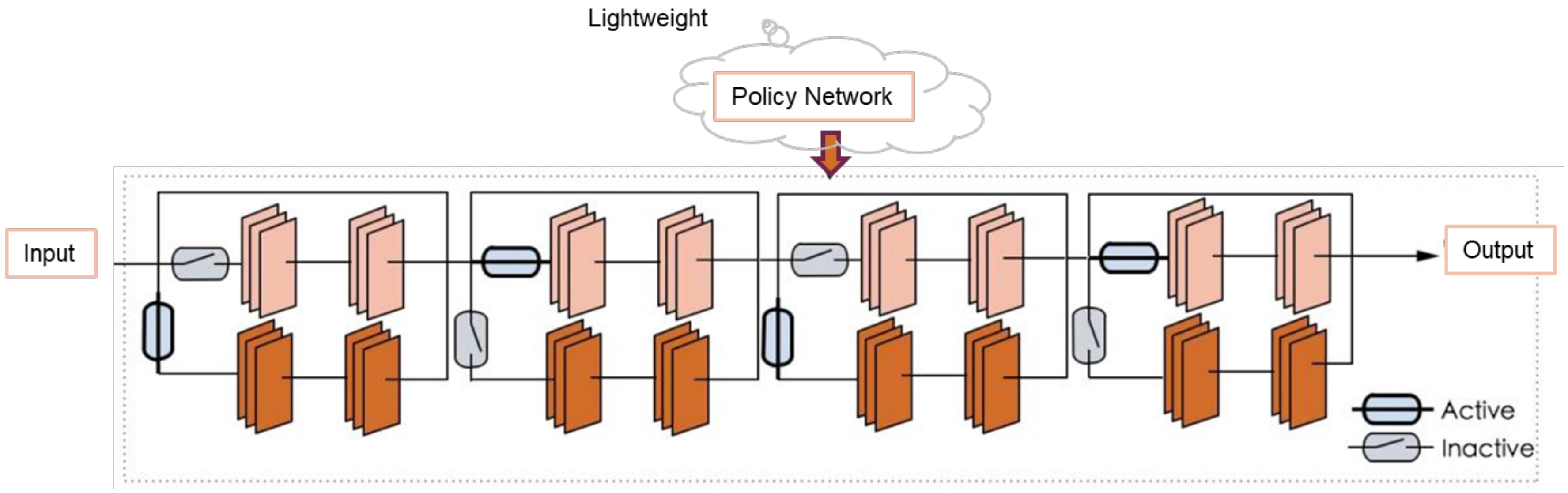


Future Work: Ensembling

- **Multiple sources may cover different interactions in target set**
  - Ensembling models may be more economic than retraining any one
  - Cost/Benefit in combination of sources to ensemble
  - Possible approaches:
    - » Choose smallest number of source sets maximizing coverage of target set
    - » Choose sets maximizing coverage of target set while minimizing source set intersection



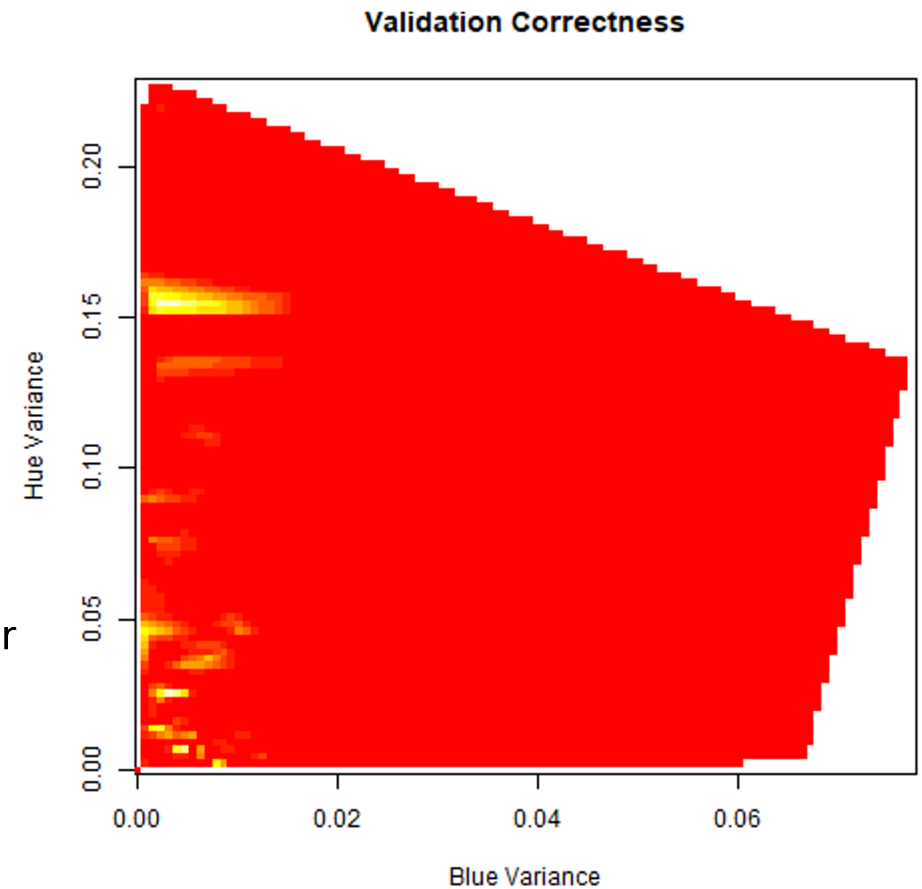
Future Work: Adaptive Fine Tuning



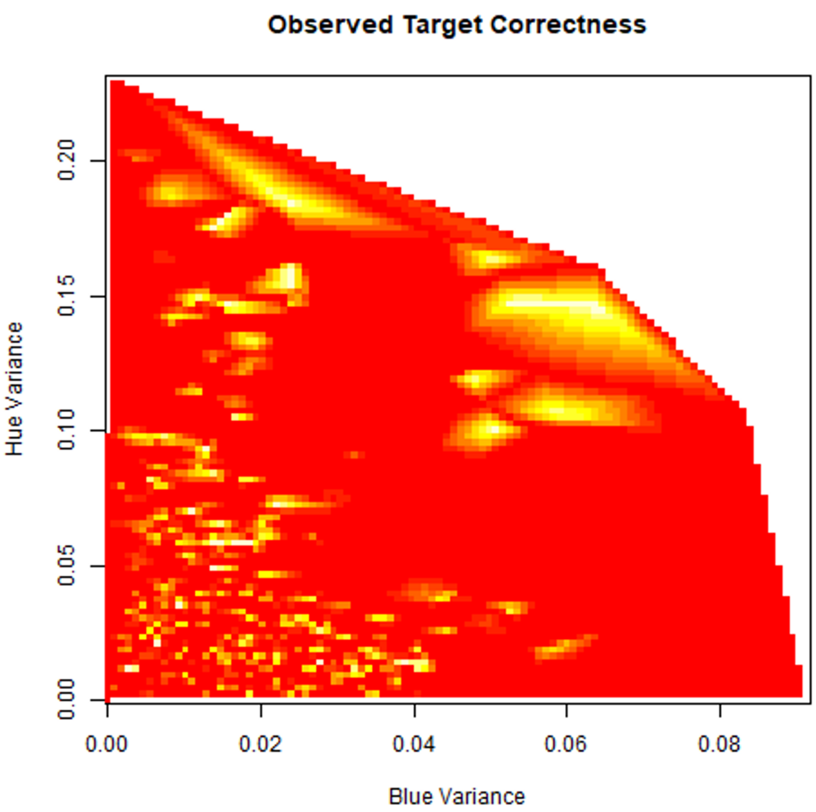
- **Approach - Develop Adaptive Fine-tuning strategy to meet desired metrics for target**
  - Identifying Pre-trained model layers that need to be fine-tuned
  - Identifying pre-trained model layers whose parameters should be frozen (shared with the source task) during training

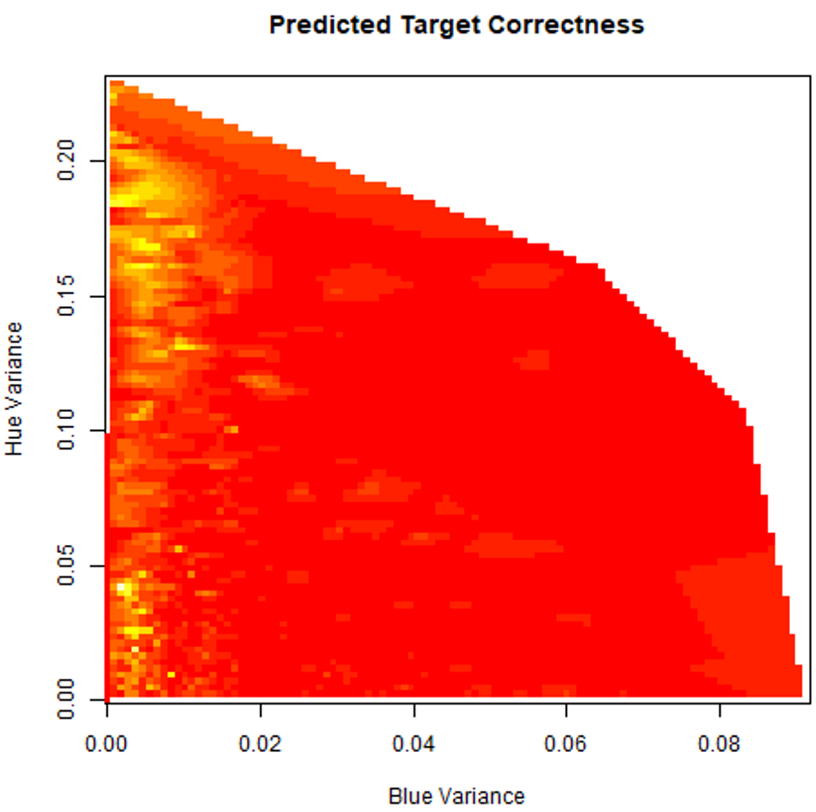
Modelling Classifier Performance

- **Analysis Goals:**
  - Identify which metadata factors are related to model generalization using the validation set
  - Develop bins that will be related to performance on the target dataset.
- **Correctness = Abs (Prediction – Class)**
  - Lower is better
  - Allows us to see not only if the model classified correctly, but how confident the classification was.
  - In addition to incorrect classification, ar unconfident classification indicates an area where the model might be underperforming.

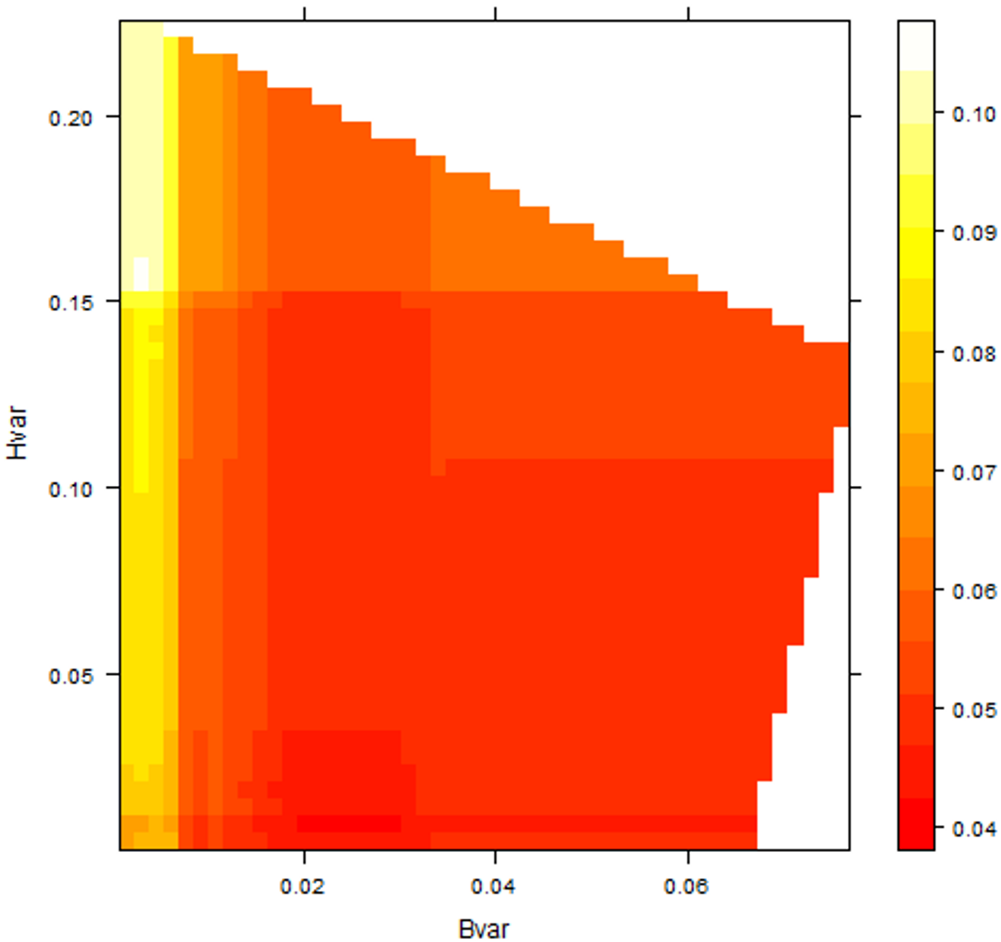


Modelling Classifier Performance





Modelling Classifier Performance



- By modelling the correctness using the validation set using a flexible model without distributional assumptions, we can get a general very loose idea of how the model will perform on the target dataset.