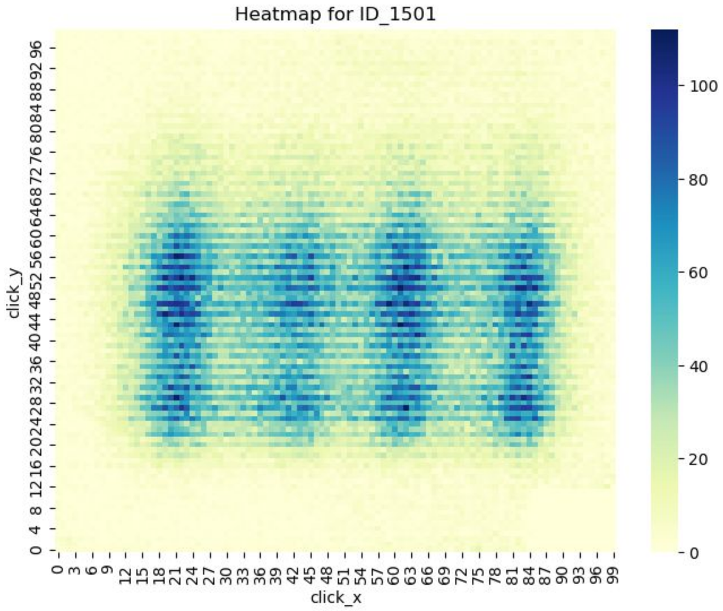
Capstone Project: Heatmap Anomaly Detection

Week 10 Progress Report

This week:

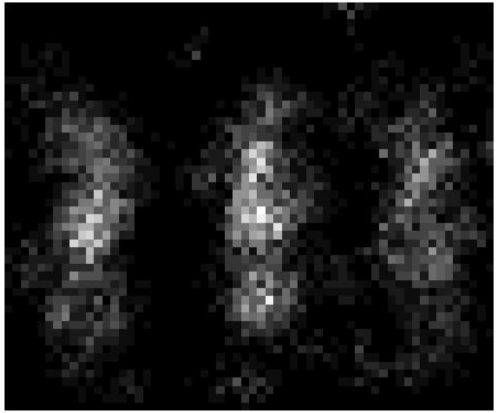
1. (from last week) Fine-tuned ResNet on different tasks
    a. Applied to new data
2. Test models on additional dataset

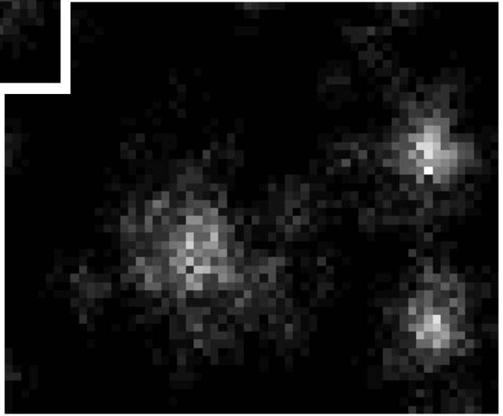
Questions



- New dataset:
  - 2 grid_ids: One seems to only have 4 products?
  - Confirmation: click_x_rel/click_y_rel = 40x40 binned?
    - Any particular reason?
- Clarifications about scale:
  - “25 M banners/day”:
    - How many new clicks/banner/day?
    - How is the data formatted?
- Compute power seems quite low (esp 8GB memory)?
  - How much time could we use / day?
  - Problematic even for inference.
- Confirmation:
  - We have 5 grid_id’s
  - The triple (grid_id, #products, domain) =: banner?

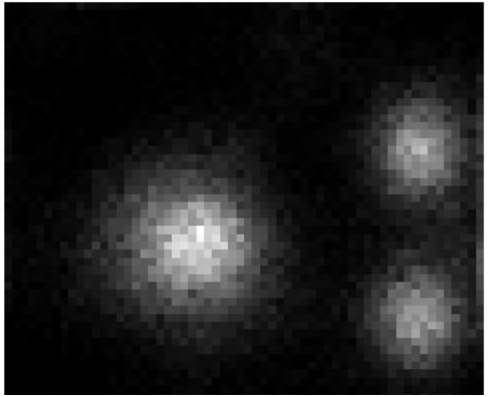
Pretrained ViT/ResNet (recap):

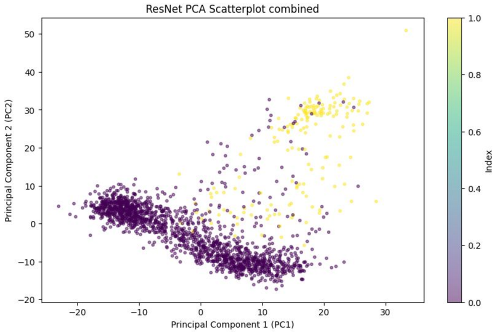




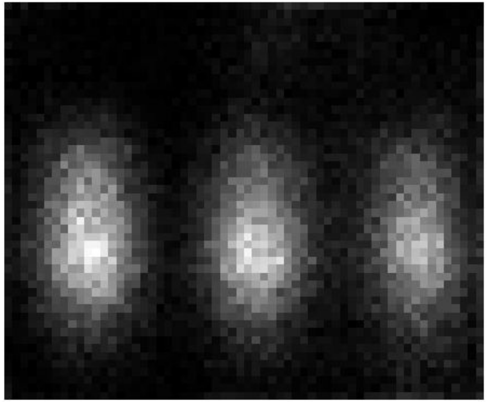
- Feed (transformed and binned) heatmaps into pre-trained ViT/ResNeT.
  - google/ViT: Transformer-based architecture, 14M images (224x224), 21k classes
  - Microsoft/ResNet-1k: trained on ImageNet-1k (224x224), 1k classes.
- Extract features (before classification head)
  - ViT → 151296 dim'l feature vector
  - ResNet → 2048 dim'l feature vector
- Play with upsampling (bootstrapping + noise)
- PCA and other dim'l reduction techniques
  - Apply clustering methods

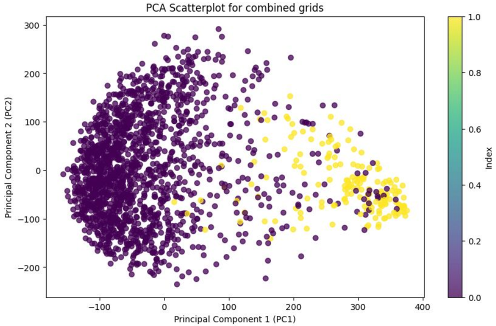
Data enhancement (recap):

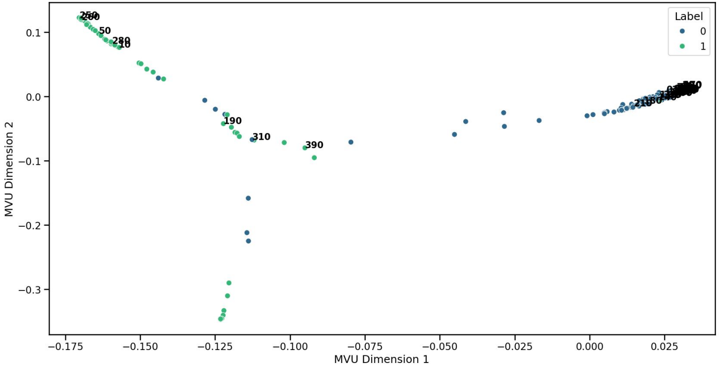




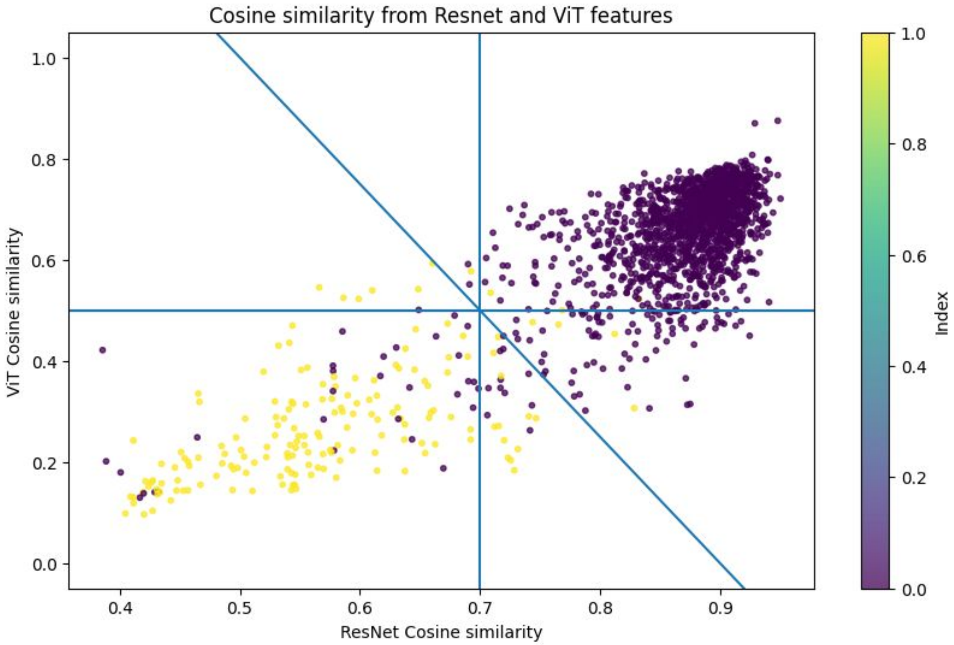
- We now bootstrap upsample to 50’000 clicks per image.
- PCA’s look a lot more promising.
- MVU also look very indicative of deep structure from pre-trained models.
- => continue with enhanced data from now on.





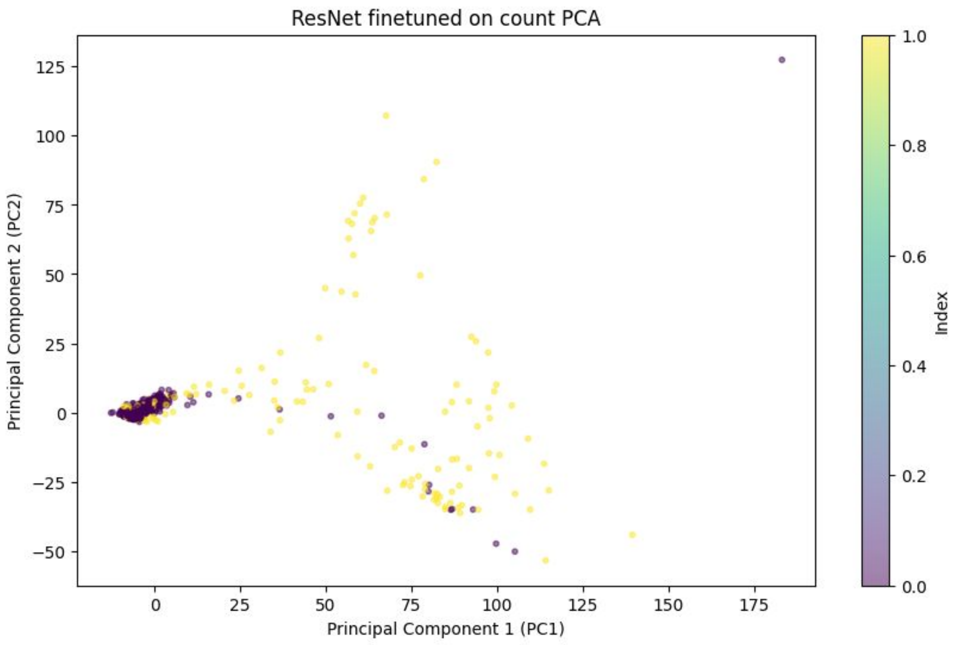


Cosine similarity (recap)



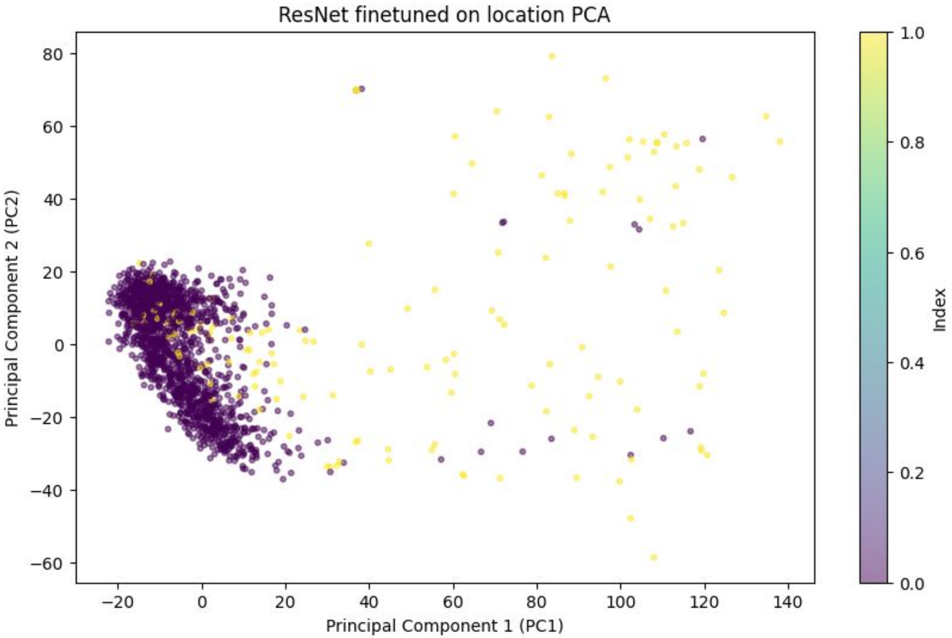
- PCA → gives some intuition, but very limited
- MVU → shows there is underlying structure.
- Feature vectors usually evaluated using **cosine similarity** (angle between vectors in high-dimensional feature space)
- **Cosine similarity** has actual meaning (while PCA does not really, since "maximum variance" is inherently imprecise).
- Draw picture of cosine similarity using ResNet and ViT feature vectors.
- What choice of "anker"?
    - Pick random choice
    - Pick average non-broken grid_id's.

Fine-tune ResNet-50 on number of clusters



- Fine-tune feature vectors of ResNet-50 architecture:
  - Pretrained ResNet
  - Generate synthetic images:
    - Random # of clusters (0-20)
    - Random center for clusters
    - Random covariance for Gaussians
    - Random # of clicks/cluster
    - Overlay random noise
  - Add classification head (softmax)
  - "Learn" detection of number of clusters (cross-entropy loss).
- After some training get ~ 40% accuracy on predicting # of clusters from a given image (somewhat depending on some of the hyperparameters).
- PCA combines the two clusters into one (= have the same number of clusters).

Fine-tune ResNet-50 on position of clusters

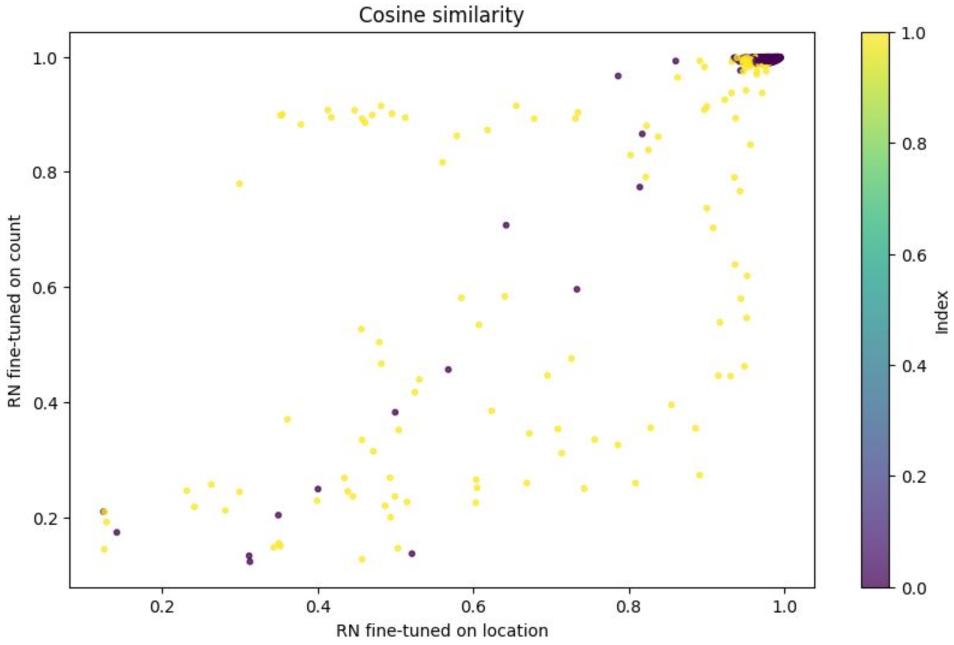


- Fine-tune feature vectors of ResNet-50 architecture:
  - Pretrained ResNet
  - Generate synthetic images:
    - Random # of clusters (0-6)
    - Random center for clusters
    - Random covariance for Gaussians
    - Random # of clicks/cluster
    - Overlay random noise
  - Add classification head providing a set of centers {(x1,y1),(x2,y2),..., (x6,y6)}
  - Setwise distance loss function; **Chamfer distance**:

$$c(A,B) = \sum_{i=1}^{|A|} \mathrm{dist}(a_i, \mathrm{nn}(B, a_i)) + \sum_{j=1}^{|B|} \mathrm{dist}(b_j, \mathrm{nn}(A, b_j))$$

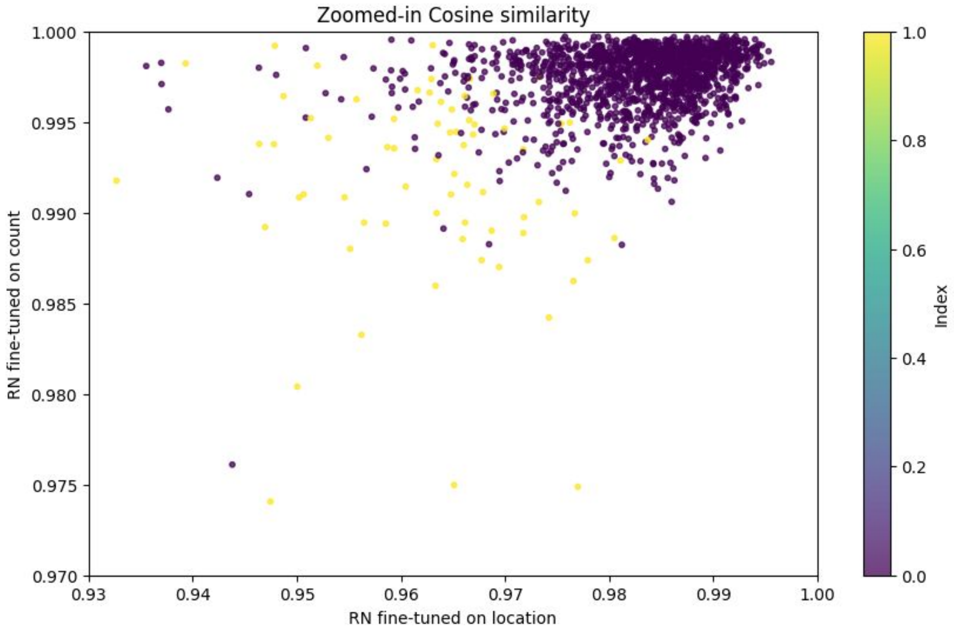
- PCA shows two clusters “pointing” in different directions (=captures difference in location).

Cosine Similarity Count vs location



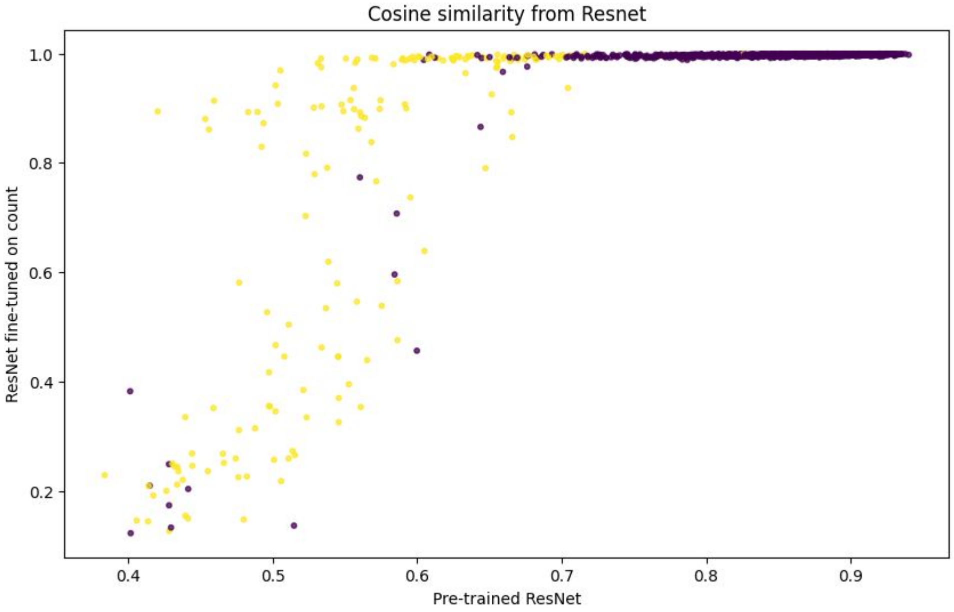
- Anker = mean of 2 unbroken clusters from each grid_id
- Cosine similarity shows a clear distinction between broken and non-broken banners
- Blue dots outside of central cluster mostly misclassified.

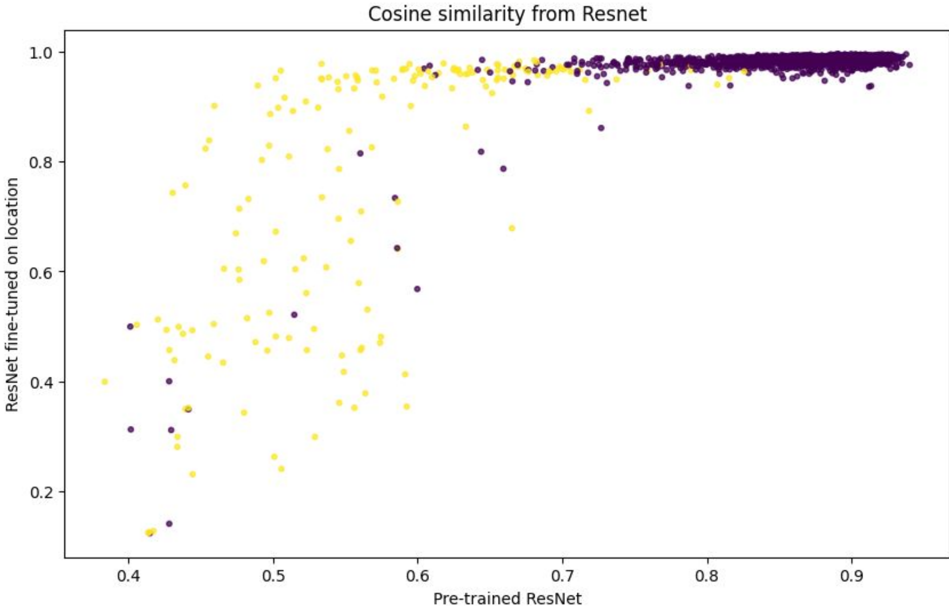
Cosine Similarity Count vs location (zoomed in)



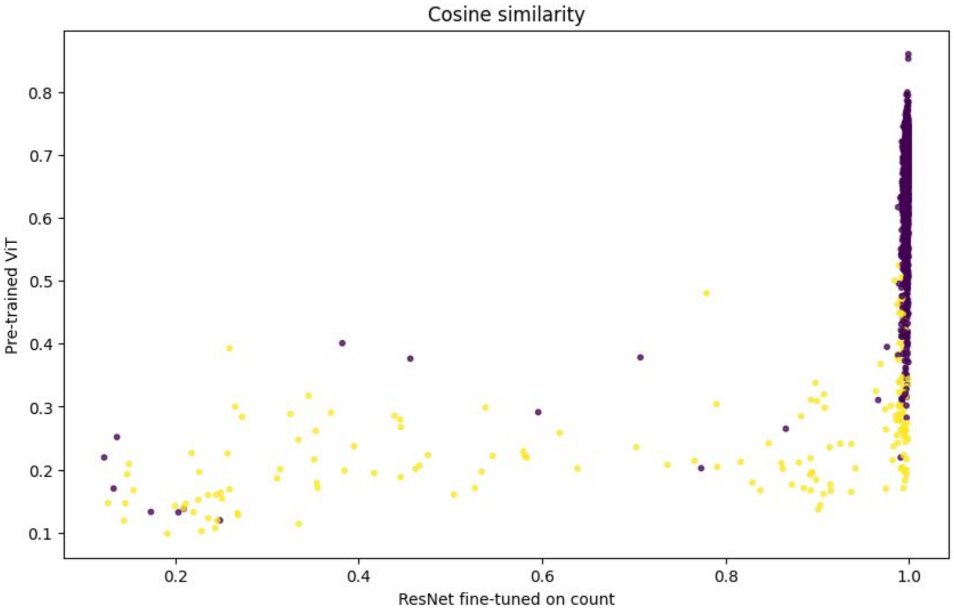
- Anker = mean of 2 unbroken clusters from each grid_id
- Cosine similarity shows a clear distinction between broken and non-broken banners
- Blue dots outside of central cluster mostly misclassified.

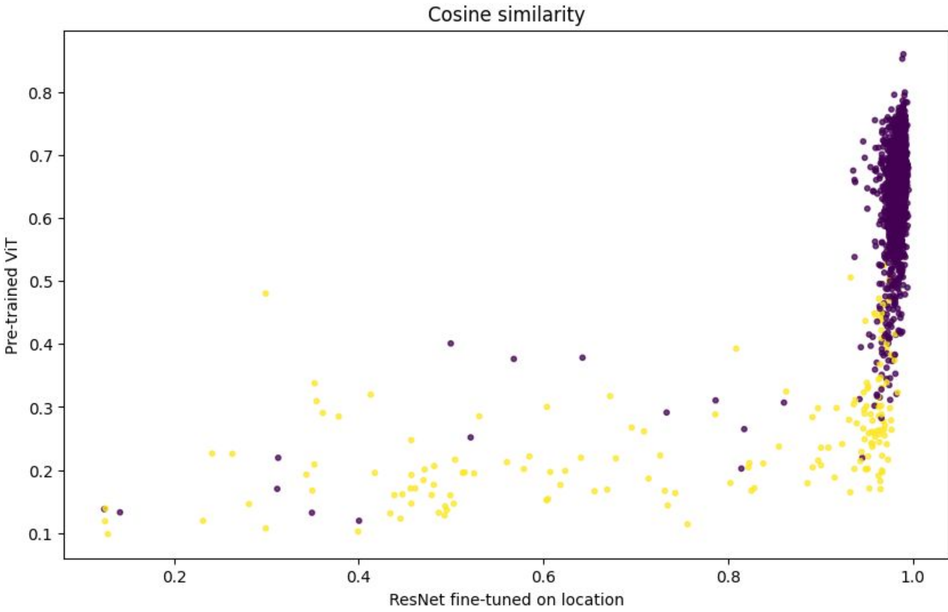
Cosine Similarity Count/loc vs Pretrained



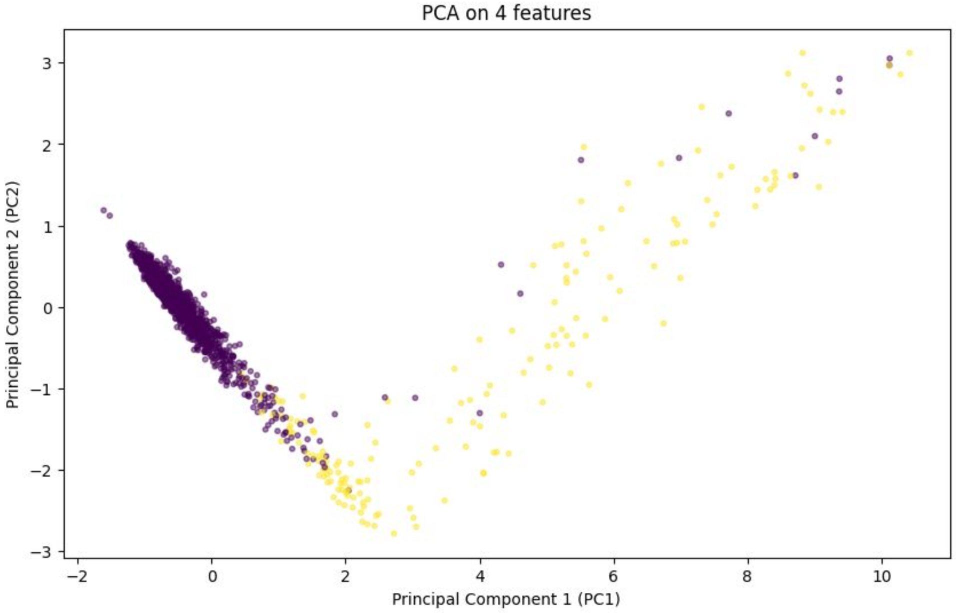


Cosine Similarity Count/loc vs ViT

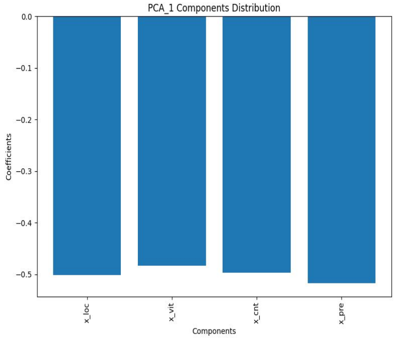


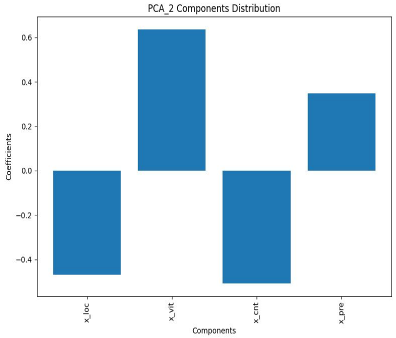


ViT, ResNet, fine-tuned Resnet

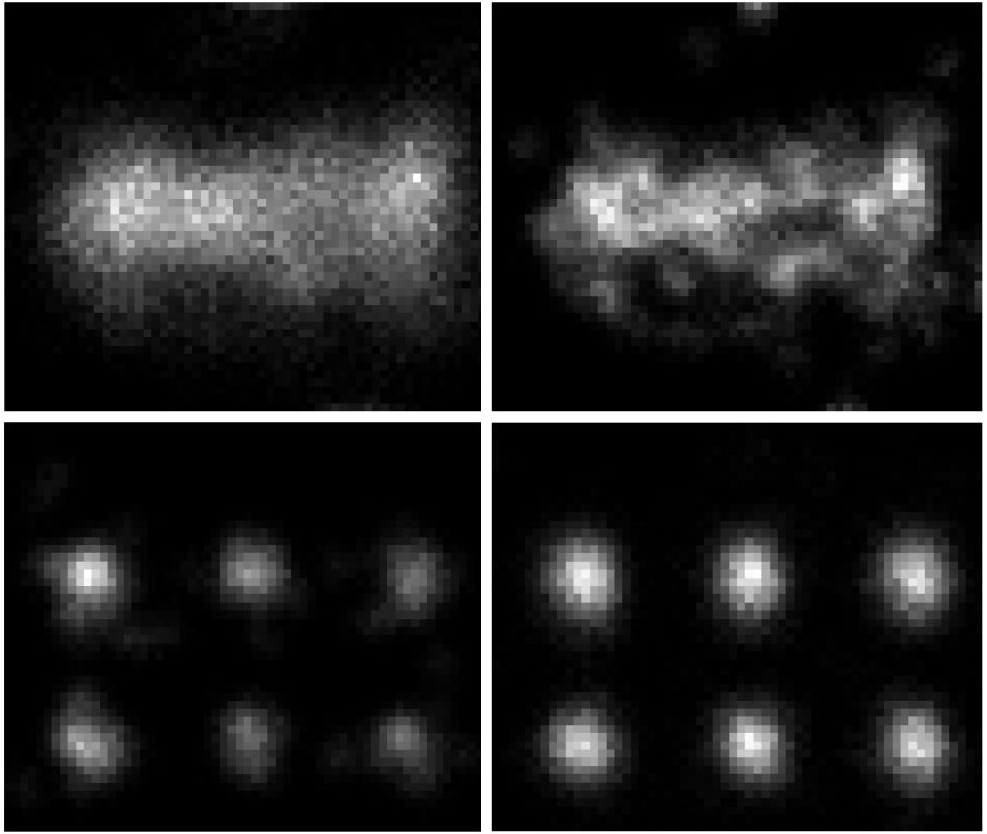


- Have now 4 powerful features (pre-trained ViT/ResNet, fine-tuned ResNet on cluster location/count).
- How do we best combine this.



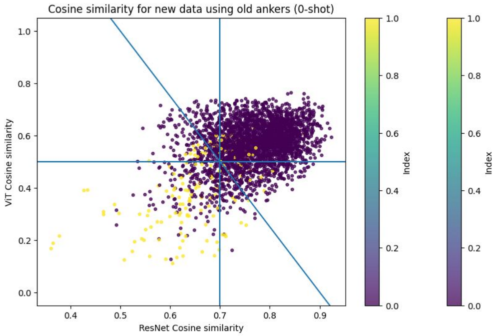


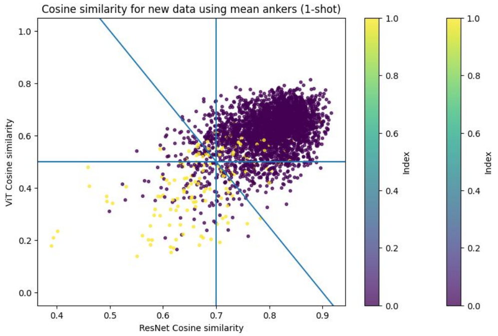
Pretrained ViT/ResNet for new data:

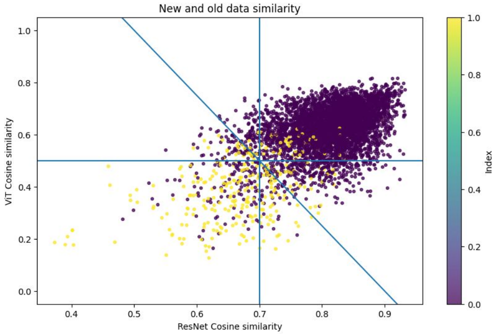


- Apply these methods for new dataset.
- Notice that we've identified 1 (at least) image as broken in previous data that looked like first image.
- Banner 333346 (top) harder to see structure → maybe change noisy upsampling?

Pretrained ViT/ResNet for new data:

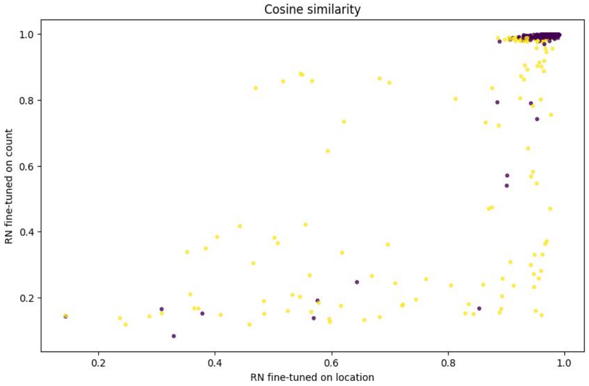


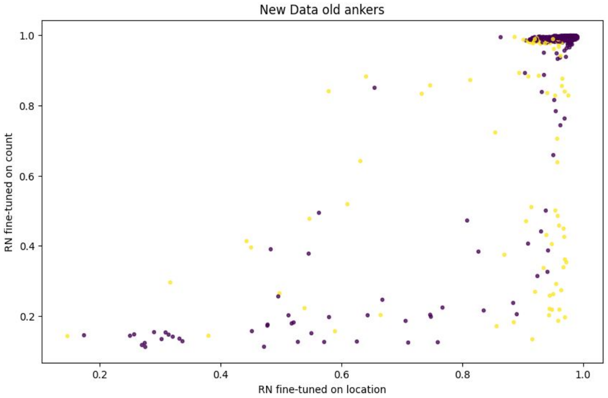


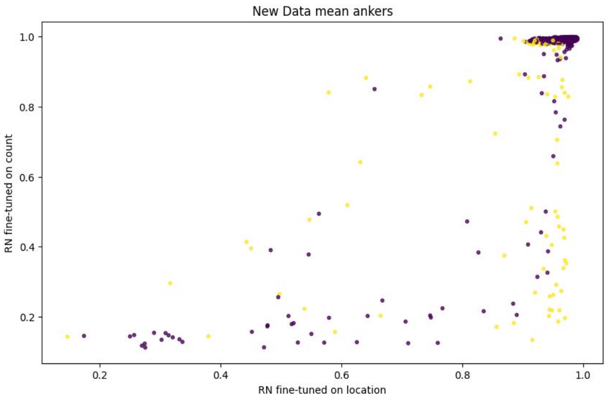


- 0-shot and 1-shot performance for new data.
  - “Old” threshold still very good
  - 1-shot better than 0-shot
- Combined data looks quite good too.
- Need to analyse some of the “misclassifications”. (to do)

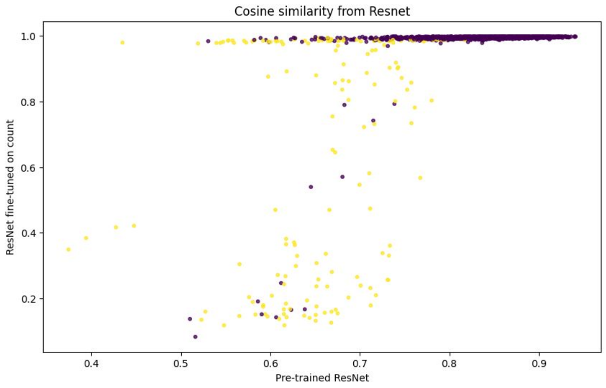
0/1-shot: count vs location

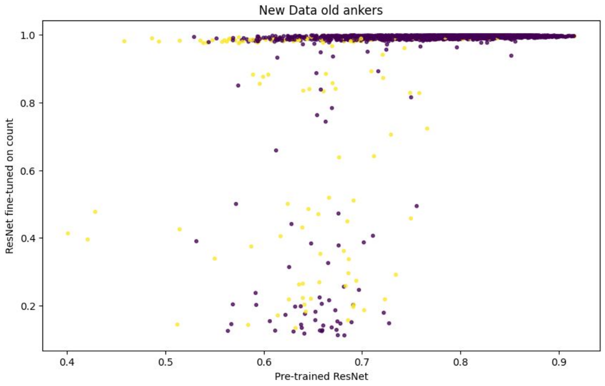


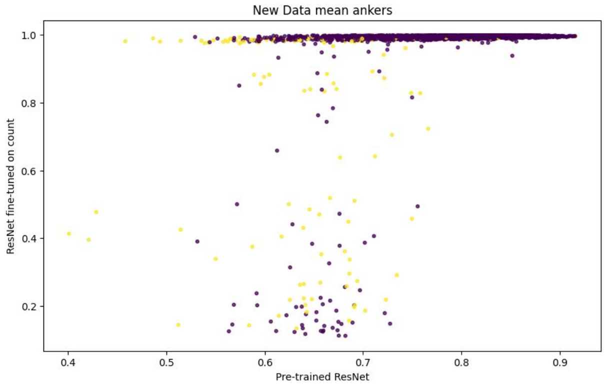




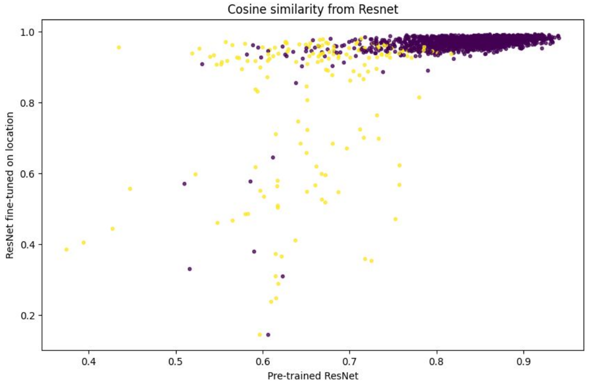
0/1-shot: count vs pretrained

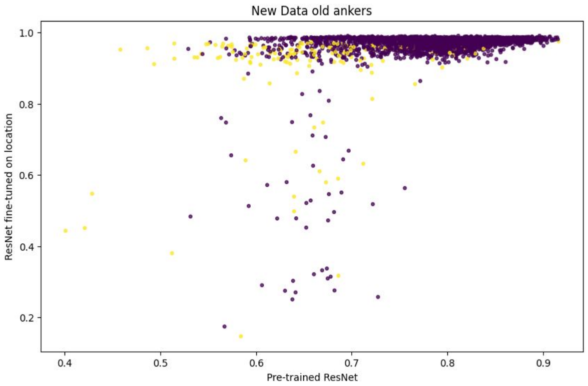


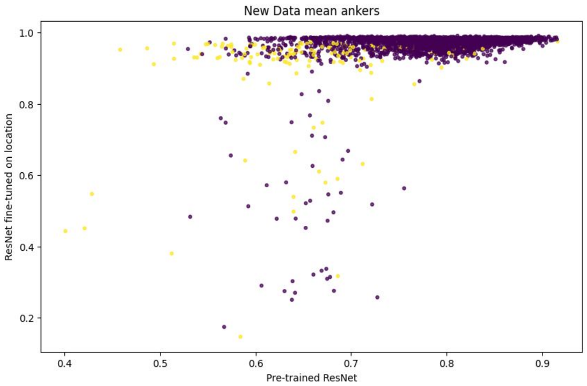




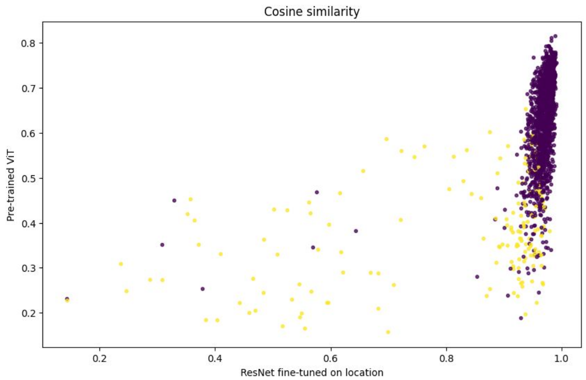
0/1-shot: location vs pretrained

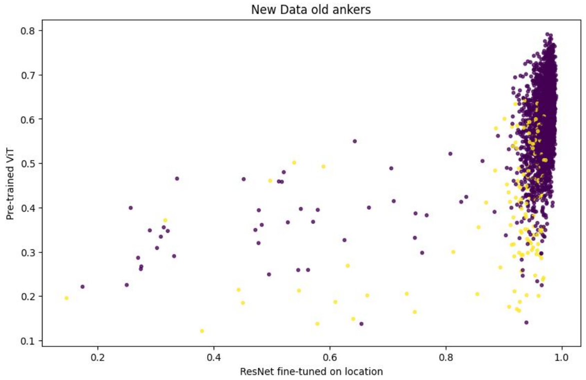


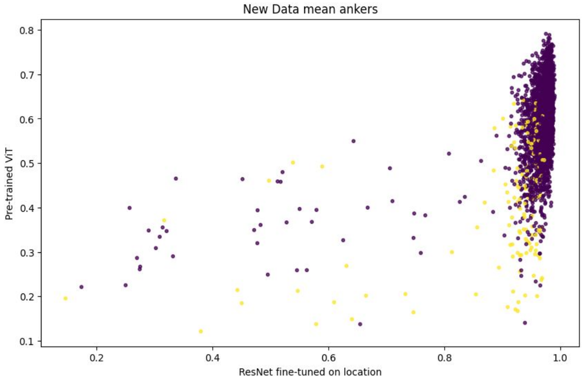




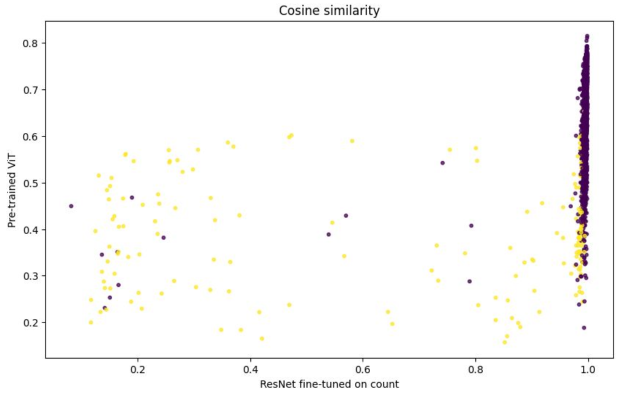
0/1-shot: ViT vs location

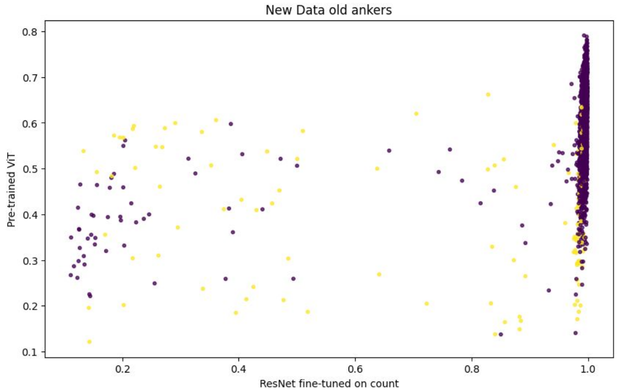


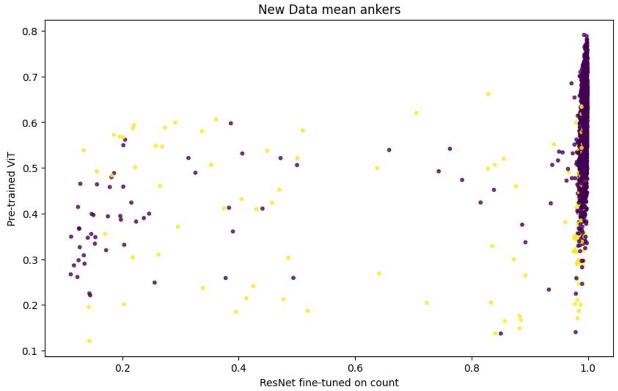




0/1-shot: ViT vs count





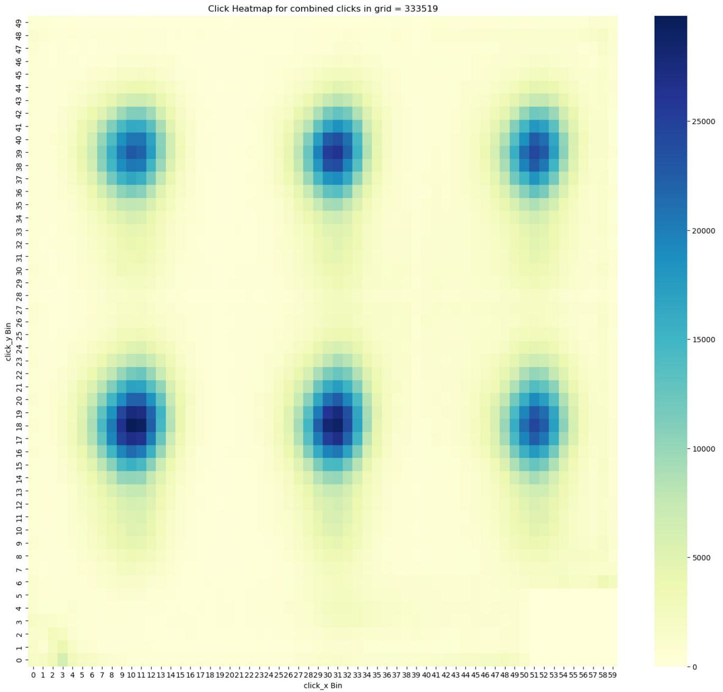


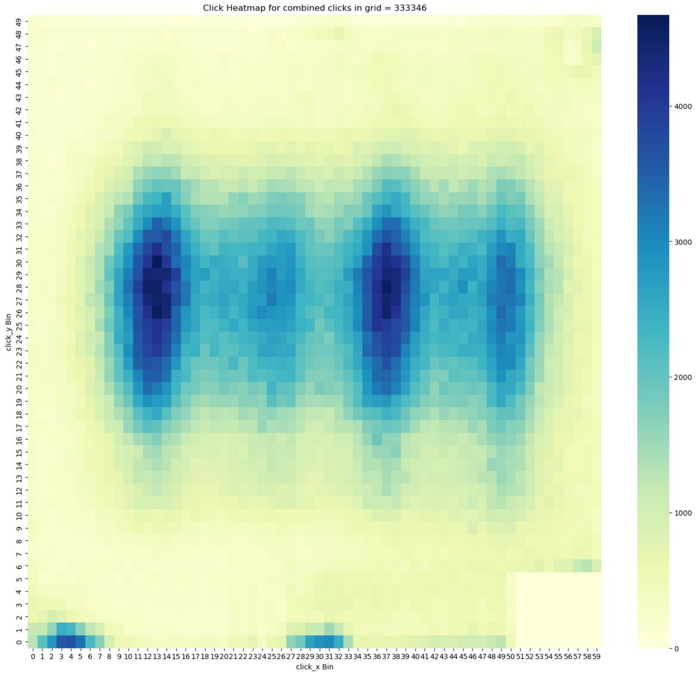
Next steps:

- Further investigate 0/1-shot performance on new dataset.
- Converge towards actual predictor
- Combine 4 ResNet/ViT features into a meaningful predictor.
    - Are there other synthetic tasks?
    - Combine with metrics features and PCA?
- SimCLR -> get it running
- Measure compute requirements more carefully:
    - Data pre-processing slow (I think can be sped up)
    - Inference times slow-ish → can be parallelized (xCPU-number speedup)

New data

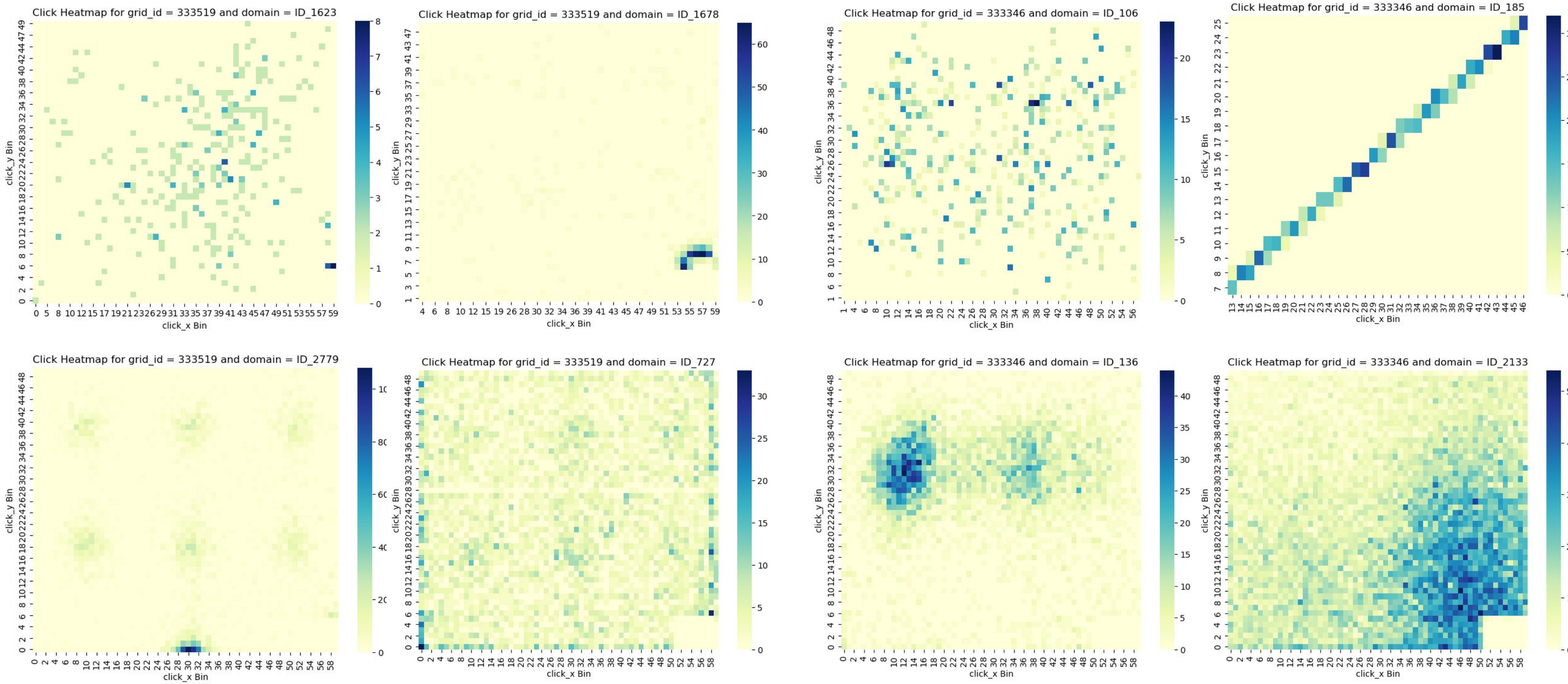
- Exploration
- Implementation





Clearly broken banners = 74/2061 = 3.59%

Clearly broken banners = 86/969 = 8.88%


Click Heatmap for grid_id = 333519 and domain = ID_1623
Click Heatmap for grid_id = 333519 and domain = ID_1678
Click Heatmap for grid_id = 333346 and domain = ID_106
Click Heatmap for grid_id = 333346 and domain = ID_185
Click Heatmap for grid_id = 333519 and domain = ID_2779
Click Heatmap for grid_id = 333519 and domain = ID_727
Click Heatmap for grid_id = 333346 and domain = ID_136
Click Heatmap for grid_id = 333346 and domain = ID_2133
click_y Bin
click_x Bin

Test on subSet of New Data

- Use our trained models from previous data
- Predict on subset of the new data (796)
- 550(333519) & 246(333346)

Predicted Result from previous combined model:

- Anomaly Percentage: 7.79%
- Number of Broken Banner Predicted: 62

On new data

SVM:
F1 Score: 0.14

KNN:
F1 Score: 0.39

K-Means:
F1 score: 0.02

DBScan:
F1 Score: 0.49

Isolation Forest:
F1 Score: 0.37

Combined Model:
F1 Score: 0.40

On previous data(test set)

SVM:
F1 Score: 0.25

KNN:
F1 Score: 0.83

K-Means:
F1 score: 0.24

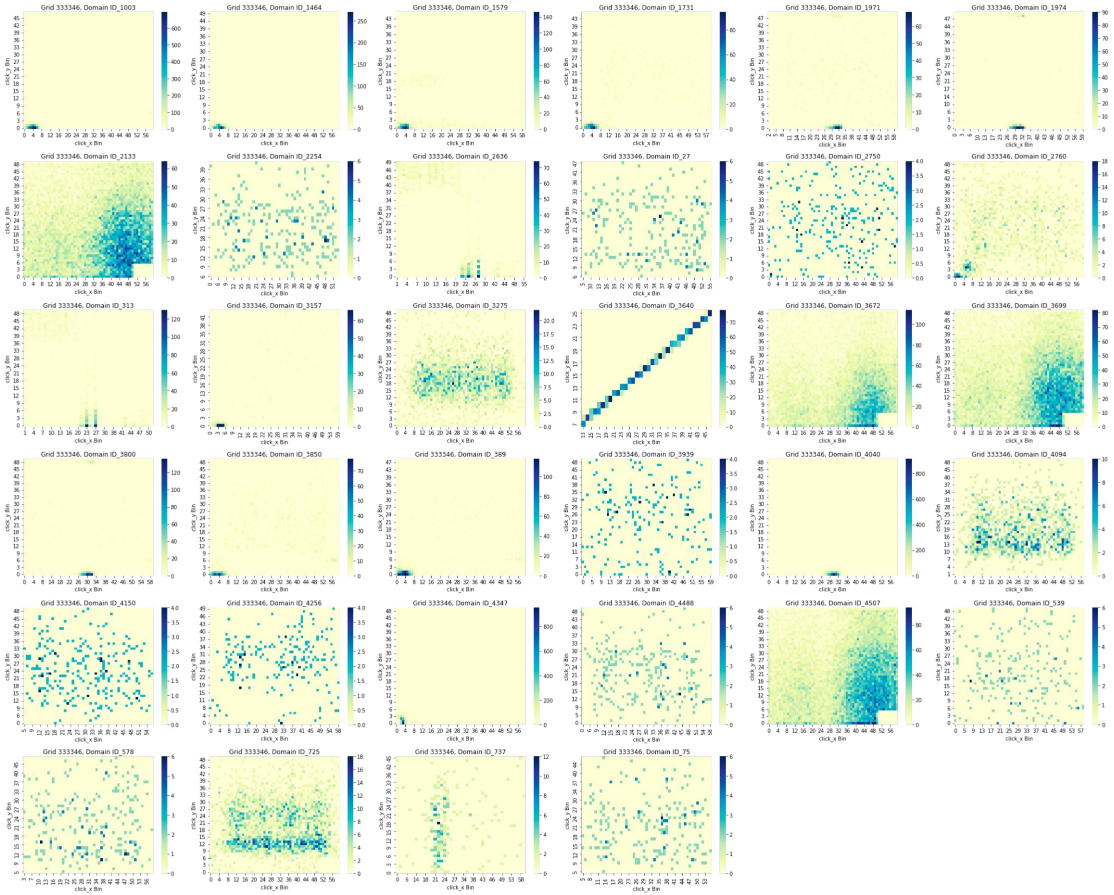
DBScan:
F1 Score: 0.72

Isolation Forest:
F1 Score: 0.85

Combined Model:
F1 Score: 0.87

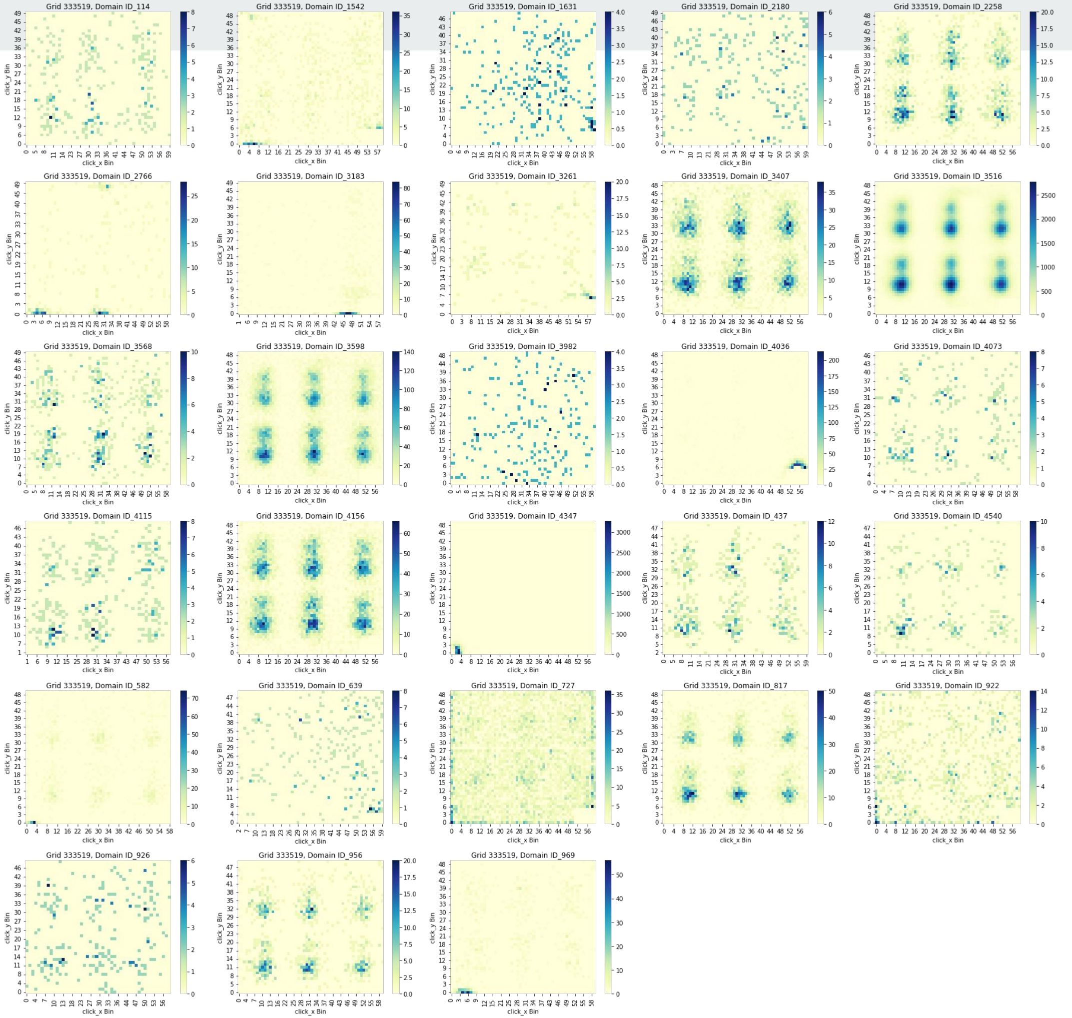
Combined Model Result1:

- This is Grid ID 333346
- For Grid ID 333346, there are 34 anomalies
- 34/246 = 0.138 → 13.8%



Combined Model Result2:

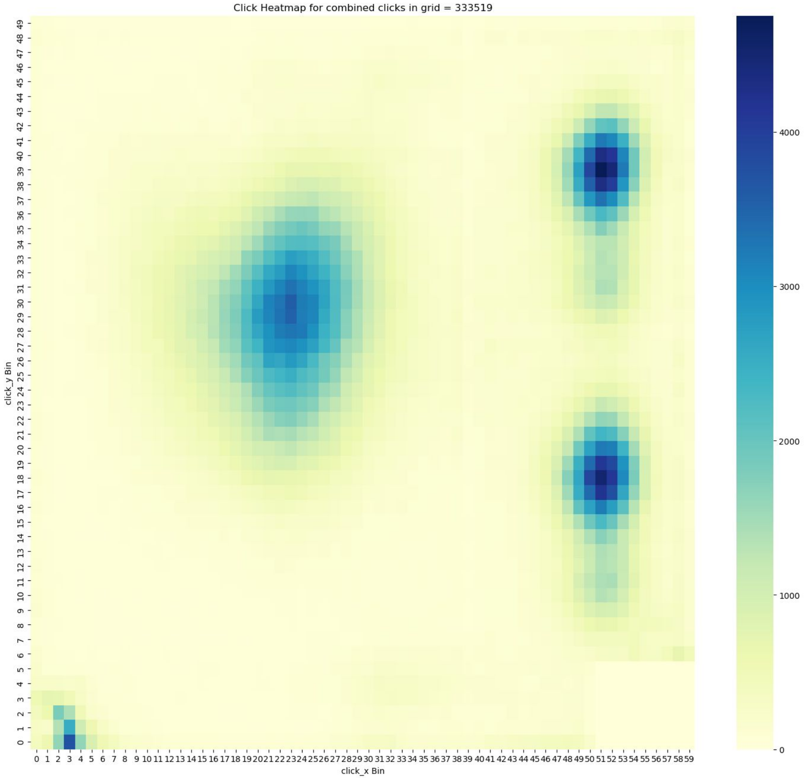
- This is Grid ID 333519
- For Grid ID 333519, there are 28 anomalies
- $28/550 = 0.051 \rightarrow 5.1\%$



Further steps:

- More exploration about the data

Old/New Statistical approach (wip):



- Cumulative clicks → empirical distribution f
- New banner → drawn from distribution
- **Draw probability:**
  - Can compute P[New banner|emp distribution]
  - Normalize: p = P[NB|ED]/P[Avg B|ED]
  - If p < threshold: broken.
    - ~30% identified as broken
- **Chi-squared:**
  - Chi-squared test:
  - Is the underlying distribution the same between ED and NB?
    - ~30% identified as broken
- **LRT:**
  - Estimate "bad banner" distribution
  - Compute p = P[NB|ED]/P[Avg B|BB]
    - Likelihood-ratio-test (to do).
- **Upshot:**
  - Lightweight and can be easily implemented on data-stream.
- **To do:**
  - Different resolutions
  - Upsample/downsample
  - "pseudo-distance" between good and bad banners (e.g. KL-divergence)

Appendix