Capstone Project: Heatmap Anomaly Detection

Week 9 Progress Report

This week:

1. Follow-up questions/clarifications
2. Progress on implementing a new self-supervised learning method: contrastive learning
3. Fine-tuned ResNet on different tasks

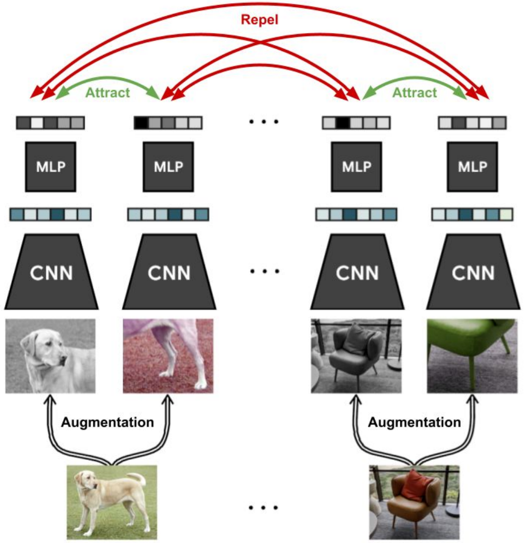
Questions

1. Regarding broken banners: question regarding the last line?
   "Our original understanding was that domain refers to a specific url on which a certain banner is displayed (independent of the specific user who looks at the webpage). However, in our discussion today, it came about that a user accessing a webpage will get personalized ads"
   - These are both true! The personalization of an ad comes after the bid is won (by Criteo) and before the ads loads – the product displayed and layout are determined by a product recommendation engine based on user preferences and performance. The final ad shown has a grid id, products shown, and performance metrics (clicks, visits, etc)
   - A single domain may see many users and ads, which is why we're able to aggregate on domain and grid id
   - **A broken banner / ad is the same no matter the user, grid id, or domain: incorrect sizing, broken code, etc that lead to unexpected activity**
   -

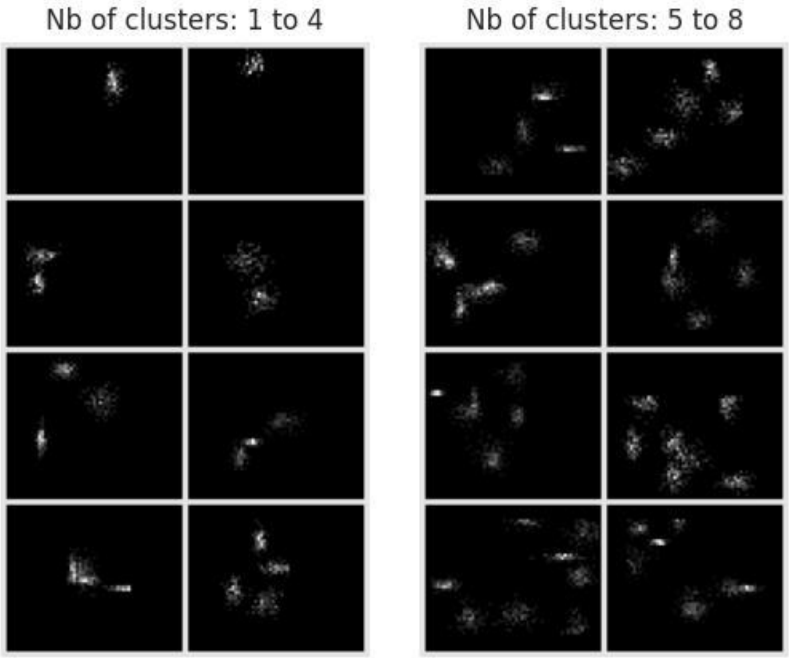
Questions

2. When will we be able to get access to the data for other types of grid_id?
3. (If possible but not important): What is the exact time period for the metrics dataset and the heatmap dataset, and also the region of the users/ads?

Contrastive learning: SimCLR



Progress on generating data

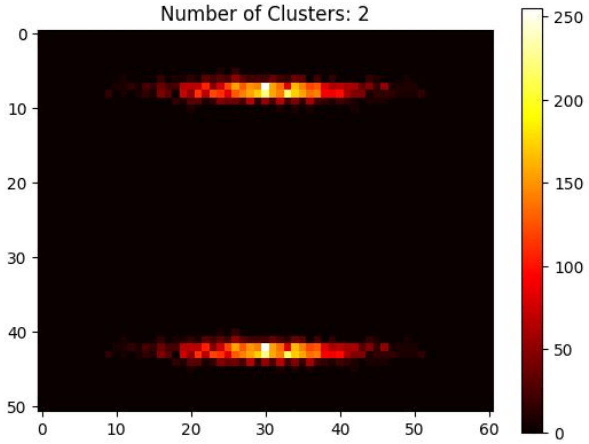


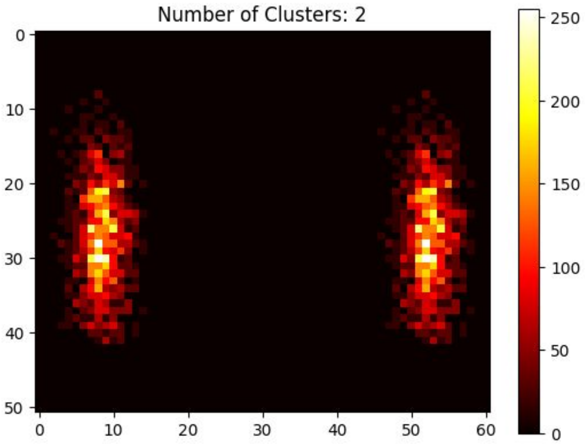
Nb of clusters: 1 to 4

Nb of clusters: 5 to 8

- Created new dataset class to generate synthetic banner data for SimCLR
- Batch size: 8 * 2, 8 classes (two instances of each class per batch needed for contrastive learning training)
- 1000 batches
- Centres randomly initialized, size of clusters too (by using random value of standard deviation in normal distribution between 1 and 10)

Next step: Generation of other types of grids to increase batch size





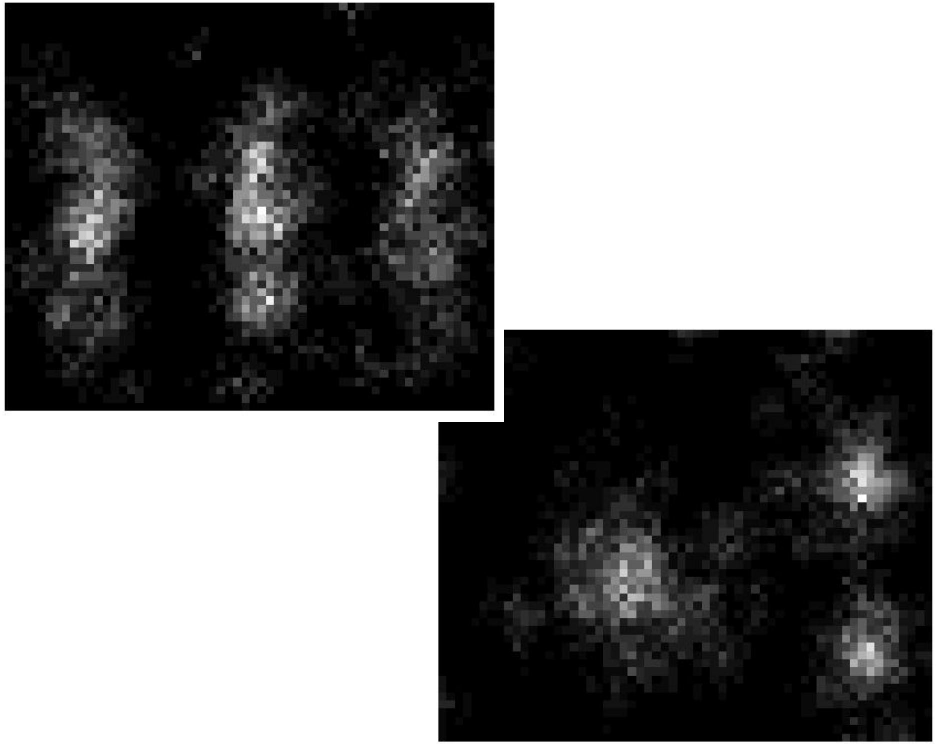
Currently in progress:

- Mix between vertical and horizontal clusters etc.
- Broken banner grids

Next steps for SimCLR

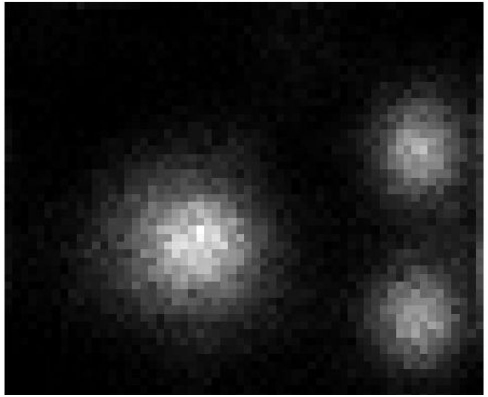
- Come up with more ideas for possible grid IDs and broken banners
  - This entails coming up with a clear definition of what is a broken banner and what are the characteristics of a grid - e.g. clicks in a corner or a line of clicks?
- Finetuning hyperparameters of the model for grid data
- Compare results against baseline ResNet model

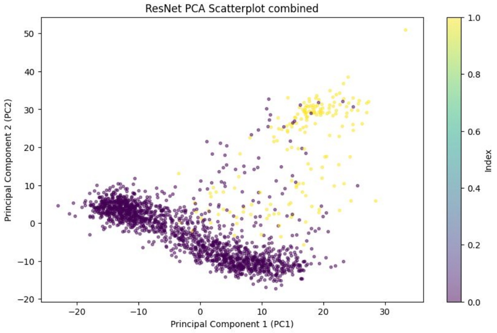
Pretrained ViT/ResNet (recap):



- 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

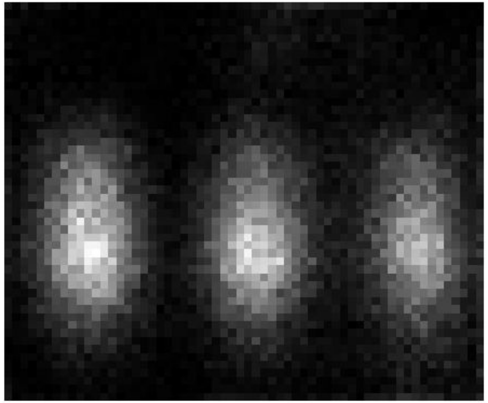
Data enhancement (recap):

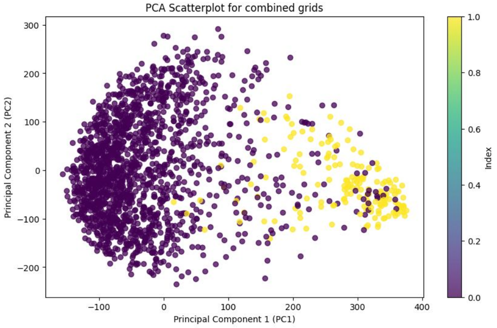




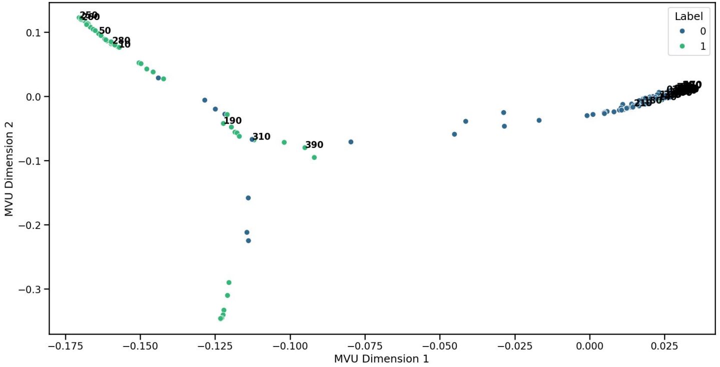
ResNet PCA Scatterplot combined

- 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.

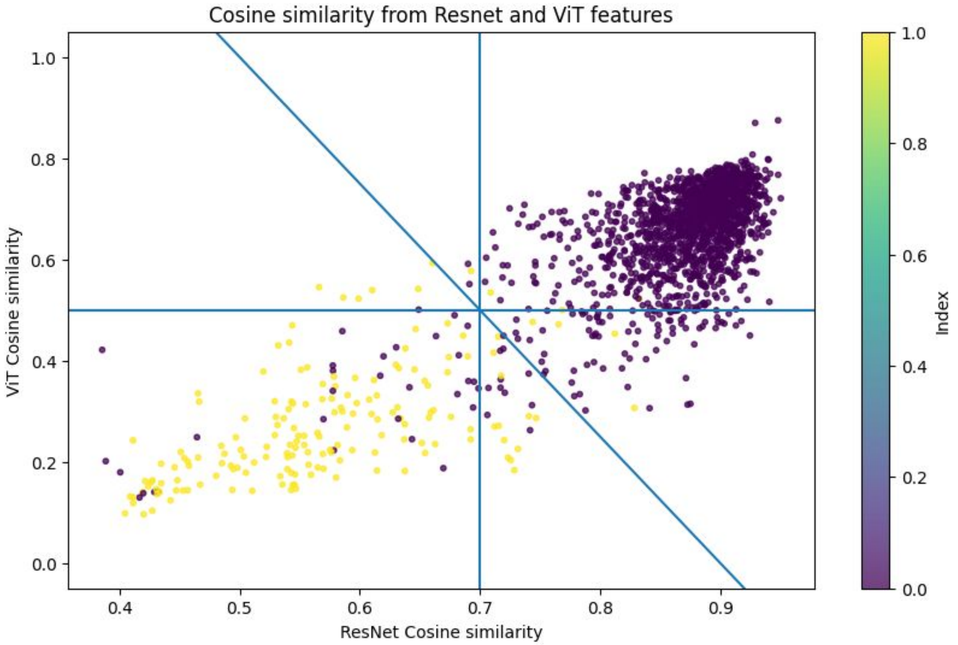




PCA Scatterplot for combined grids

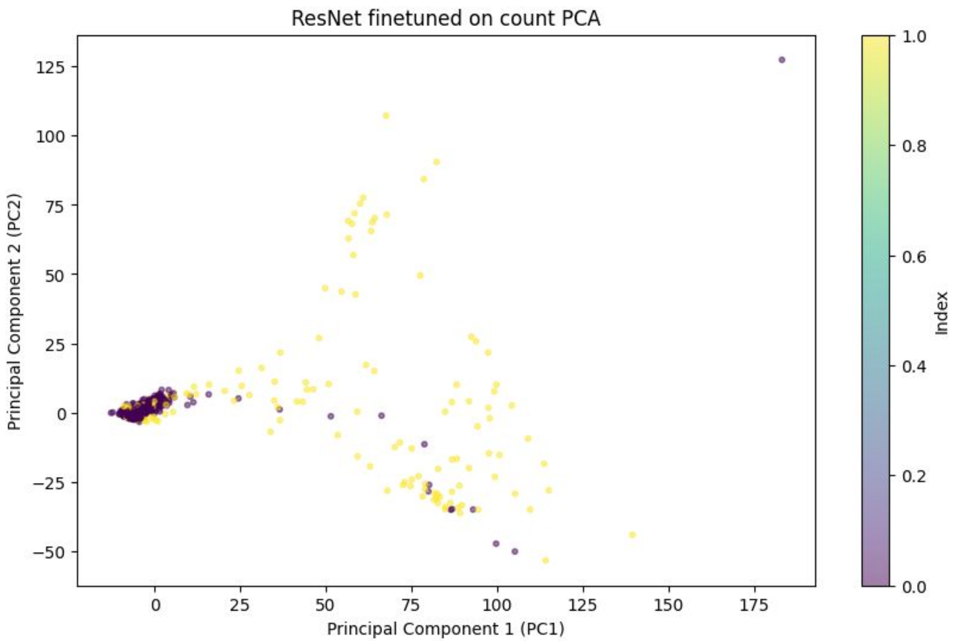


Cosine similarity (recap)



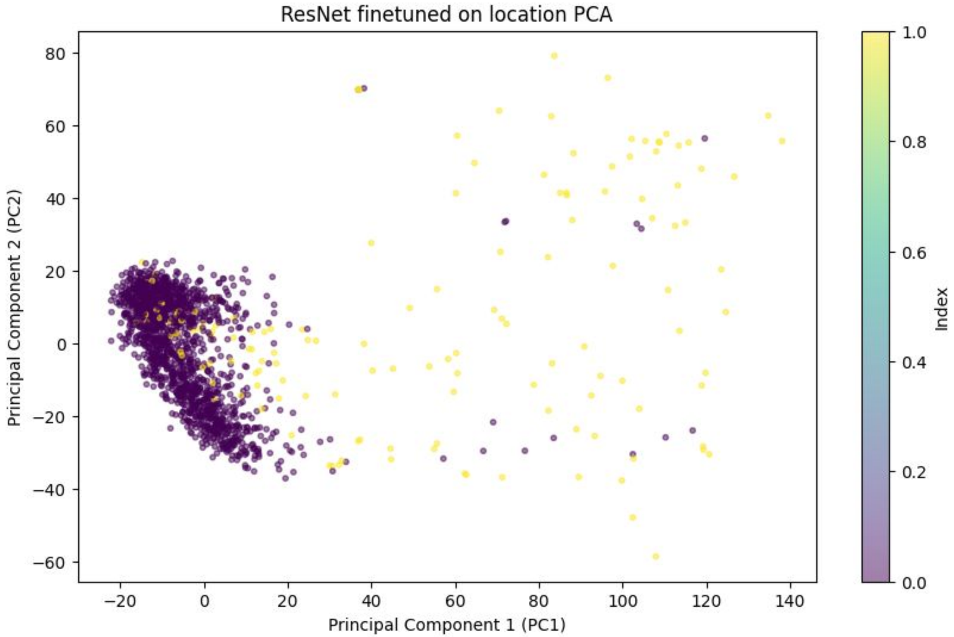
- PCA → gives some intuition, but very limited
- MVU → shows there is underlying structure.
- Feature vectors usually evaluated using **cosine similarity** (angle between vectors in high-dimensional feature space)
- **Cosine similarity** has actual meaning (while PCA does not really, since “maximum variance” 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

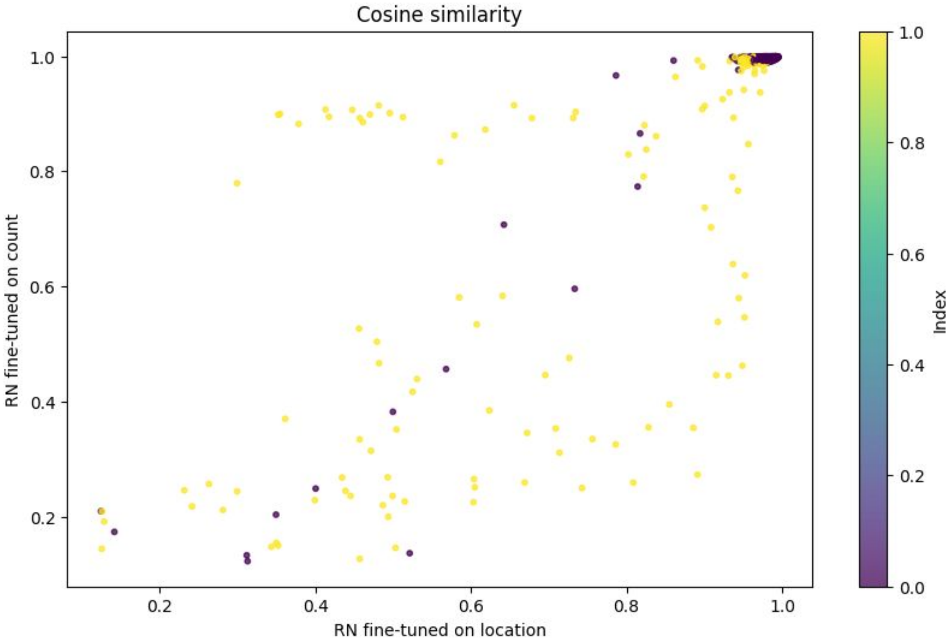


- Fine-tune feature vectors of ResNet-50 architecture:
  - Pretrained ResNet
  - Generate synthetic images:
    - Random # of clusters (0-6)
    - Random center for clusters
    - Random covariance for Gaussians
    - Random # of clicks/cluster
    - Overlay random noise
  - Add classification head providing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

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

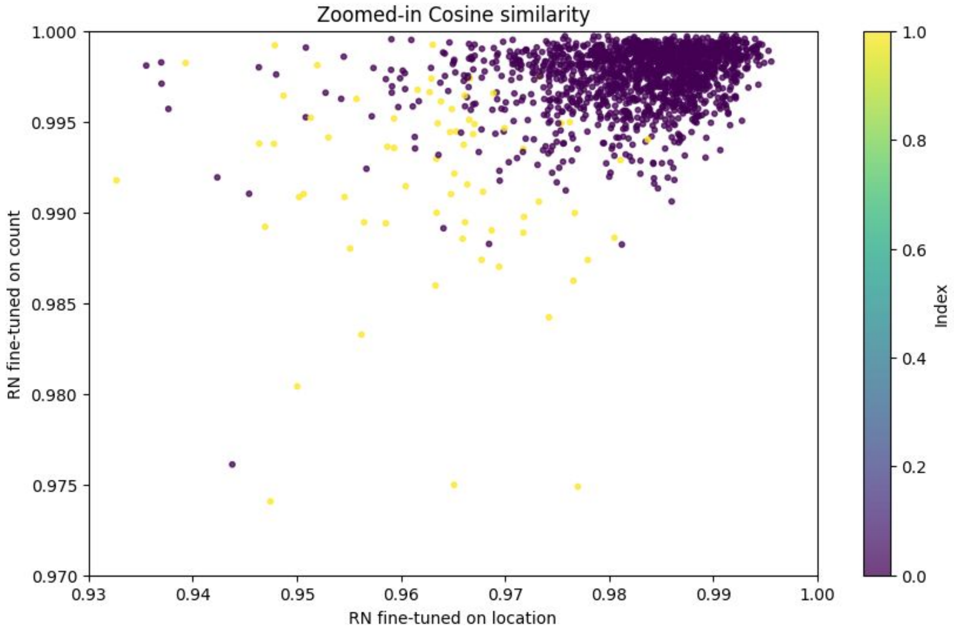
- PCA shows two clusters “pointing” in different directions (=captures difference in location).

Cosine Similarity Count vs location



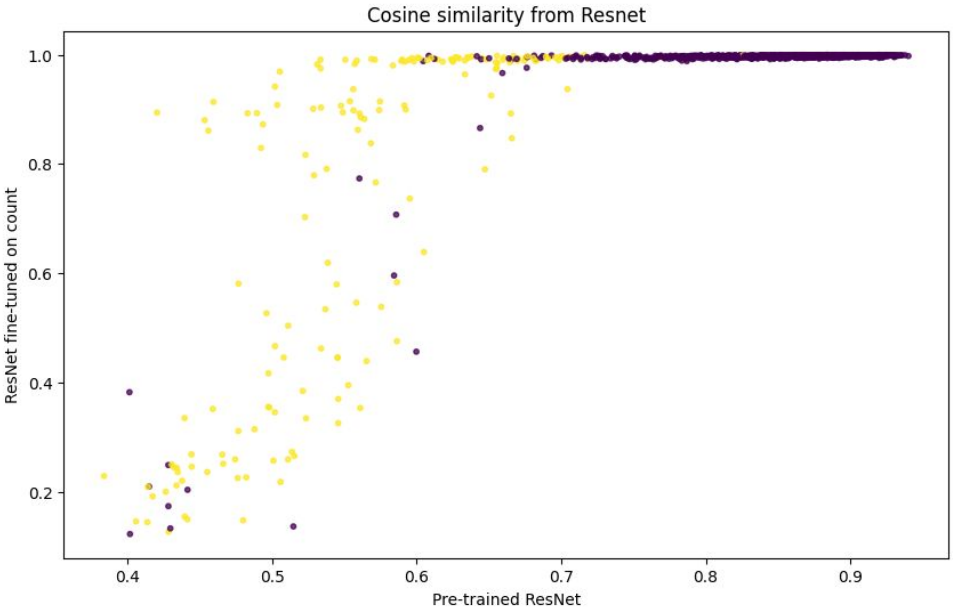
- Anker = mean of 2 unbroken clusters from each grid_id
- Cosine similarity shows a clear distinction between broken and non-broken banners
- Blue dots outside of central cluster mostly misclassified.

Cosine Similarity Count vs location (zoomed in)

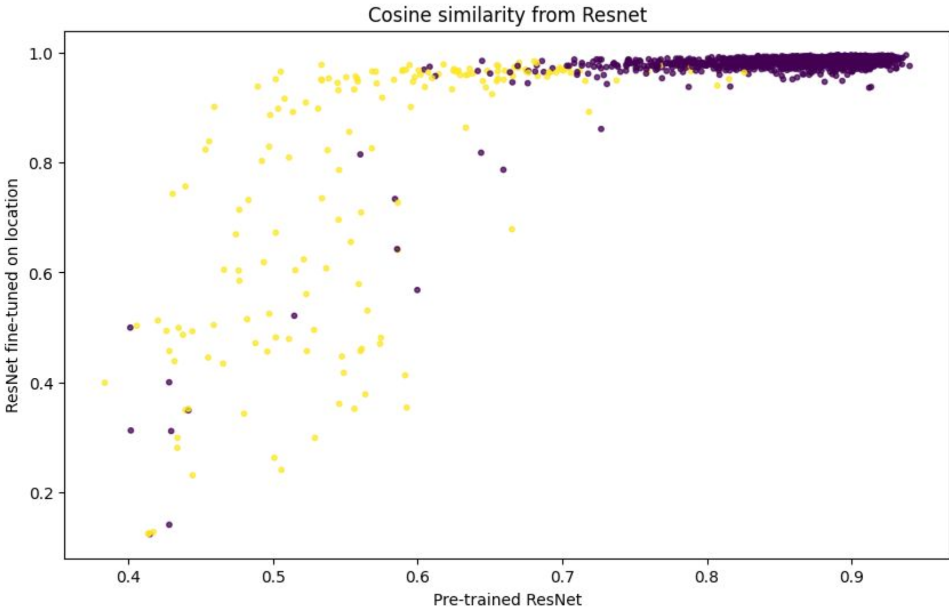


- Anker = mean of 2 unbroken clusters from each grid_id
- Cosine similarity shows a clear distinction between broken and non-broken banners
- Blue dots outside of central cluster mostly misclassified.

Cosine Similarity Count/loc vs Pretrained

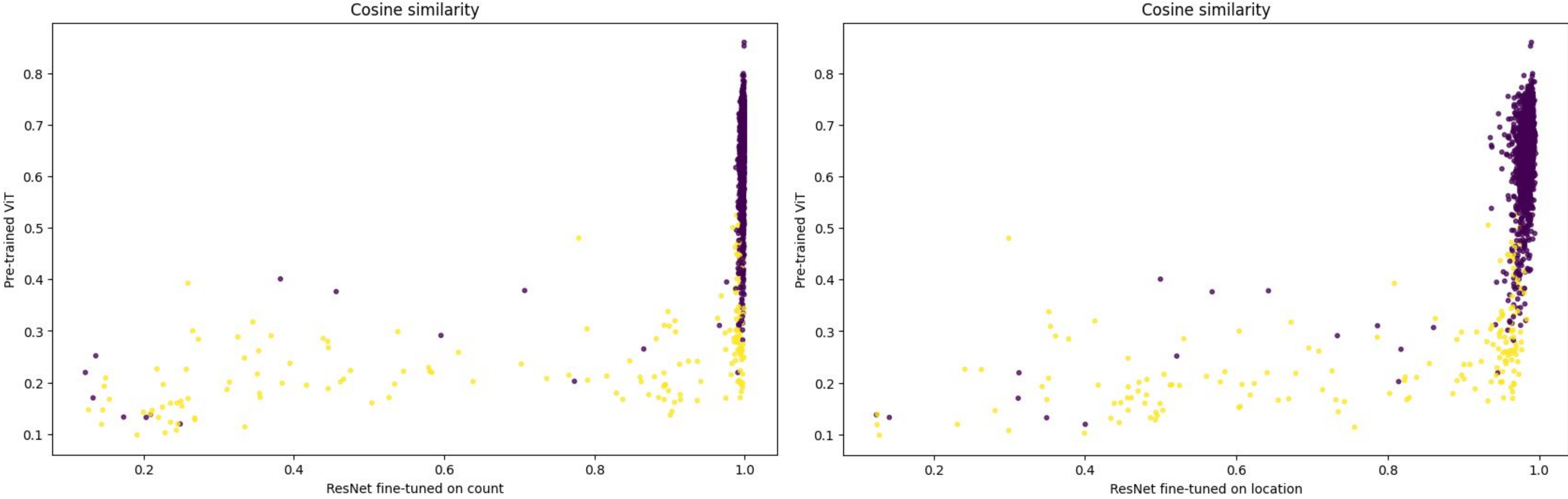


Cosine similarity from Resnet

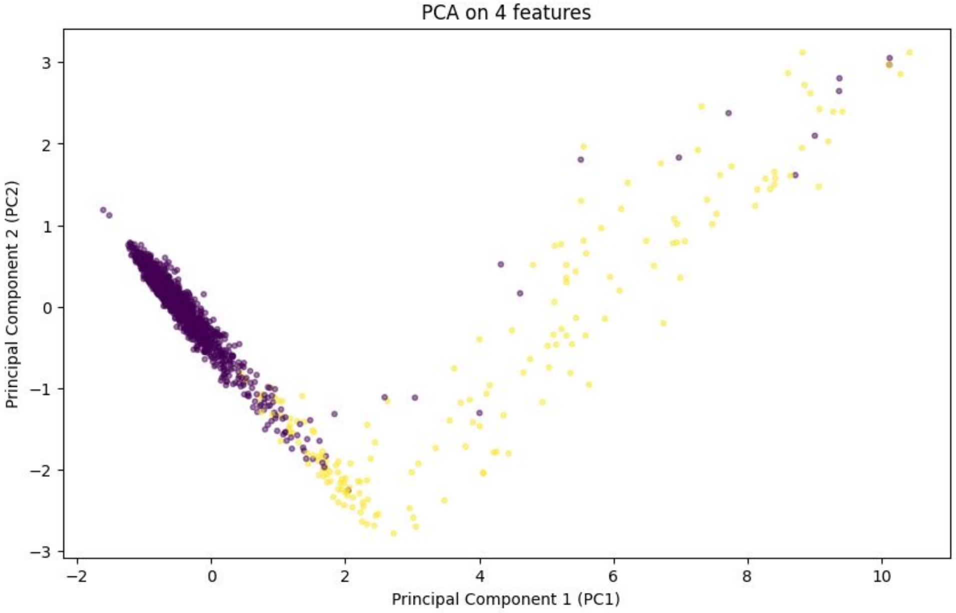


Cosine similarity from Resnet

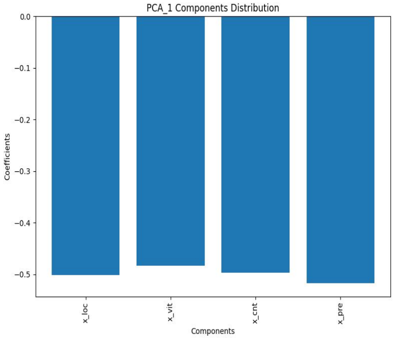
Cosine Similarity Count/loc vs ViT

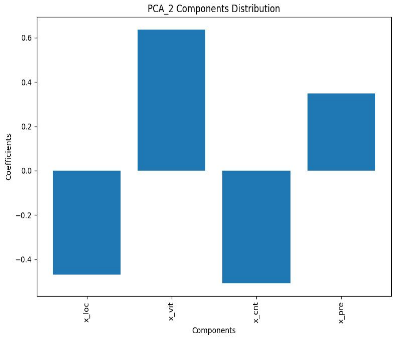


ViT, ResNet, fine-tuned Resnet



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





Next steps:

- “Click-clustering” - does it generalize?
- Combine 4 ResNet/ViT features into a meaningful predictor.
    - Are there better synthetic tasks?
    - Combine with metrics features and PCA?
- SimCLR (see above).

Appendix