Friedrich-Alexander-Universität
Erlangen-Nürnberg

Deep Anomaly Detection Using Geometric Transformations

(Golan and El-Yaniv 2018)

Final presentation - Advanced ML for Anomaly Detection WiSe 24/25

Pedro Blöss Braga[1]

[1]Friedrich-Alexander Universität Erlangen-Nürnberg, Department Mathematik

January 29, 2025

1. Introduction
2. Image Transformations
3. Normality Score
4. Feature Localization
5. Uncertainty Estimation
6. Results and Conclusion

1. Introduction
2. Image Transformations
3. Normality Score
4. Feature Localization
5. Uncertainty Estimation
6. Results and Conclusion

Intro

- (Golan and El-Yaniv 2018) "Deep Anomaly Detection using Geometric Transformations";
- Benchmark across classical datasets and models;
- Overall ROC AUC improvement;

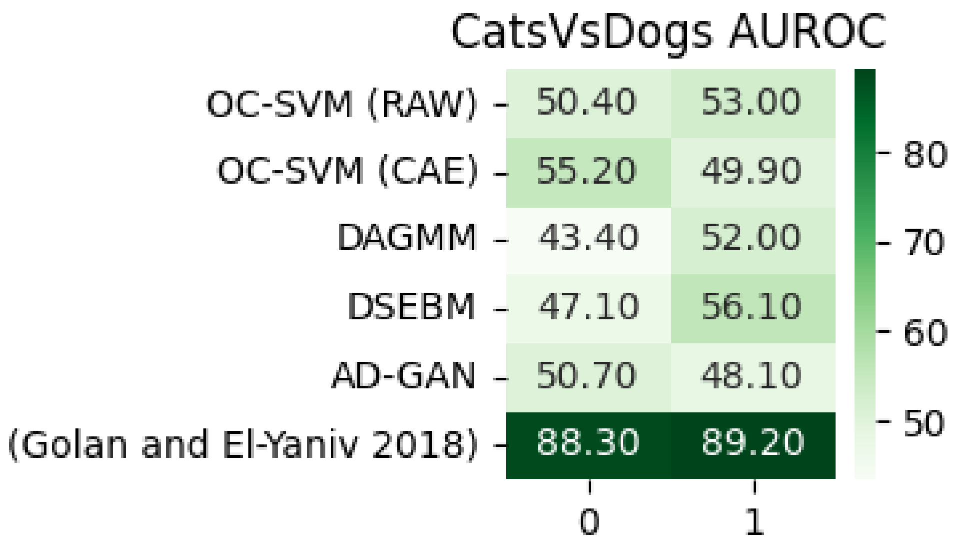


Figure: Performance benchmark with 200 Epochs.

Performance across various datasets

CIFAR-100 AUROC

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
OC-SVM (RAW)	68.00	63.10	50.40	62.70	59.70	53.50	55.90	64.40	66.70	70.10	83.00	59.70	68.70	65.00	50.70	63.50	68.30	71.70	50.20	57.50
OC-SVM (CAE)	68.40	63.60	52.00	64.70	58.20	54.90	57.20	62.90	65.60	74.10	84.10	58.00	68.50	64.60	51.20	62.80	66.60	73.70	52.80	58.40
DAGMM	43.40	49.50	66.10	52.60	56.90	52.40	55.00	52.80	53.20	42.50	52.70	46.40	42.70	45.40	57.20	48.80	54.40	36.40	52.40	50.30
DSEBM	64.00	47.90	53.70	48.40	59.70	46.60	51.70	54.80	66.70	71.20	78.30	62.70	66.80	52.60	44.00	56.80	63.10	73.00	57.70	55.50
AD-GAN	63.10	54.90	41.30	50.00	40.60	42.80	51.10	55.40	59.20	62.70	79.80	53.70	58.90	57.40	39.40	55.60	63.30	66.70	44.30	53.00
(Golan and El-Yaniv 2018)	74.70	68.50	74.00	81.00	78.40	59.10	81.80	65.00	85.50	90.60	87.60	83.90	83.20	58.00	92.10	68.30	73.50	93.80	90.70	85.00

Fashion-MNIST AUROC

	0	1	2	3	4	5	6	7	8	9
OC-SVM (RAW)	98.20	90.30	90.70	94.20	89.40	91.80	83.40	98.80	91.90	99.00
OC-SVM (CAE)	97.70	89.90	91.40	90.70	89.10	88.50	81.70	98.70	90.60	98.60
DAGMM	42.10	55.10	50.40	57.00	26.90	70.50	48.30	83.50	49.90	34.00
DSEBM	91.60	71.80	88.30	87.30	85.20	87.10	73.40	98.10	86.00	97.10
AD-GAN	89.90	81.90	87.60	91.20	86.50	89.60	74.30	97.20	89.00	97.10
(Golan and El-Yaniv 2018)	99.40	97.60	91.10	89.90	92.10	93.40	83.30	98.90	90.80	99.20

Figure: Performance benchmark with 200 Epochs.

Original Framework - Training

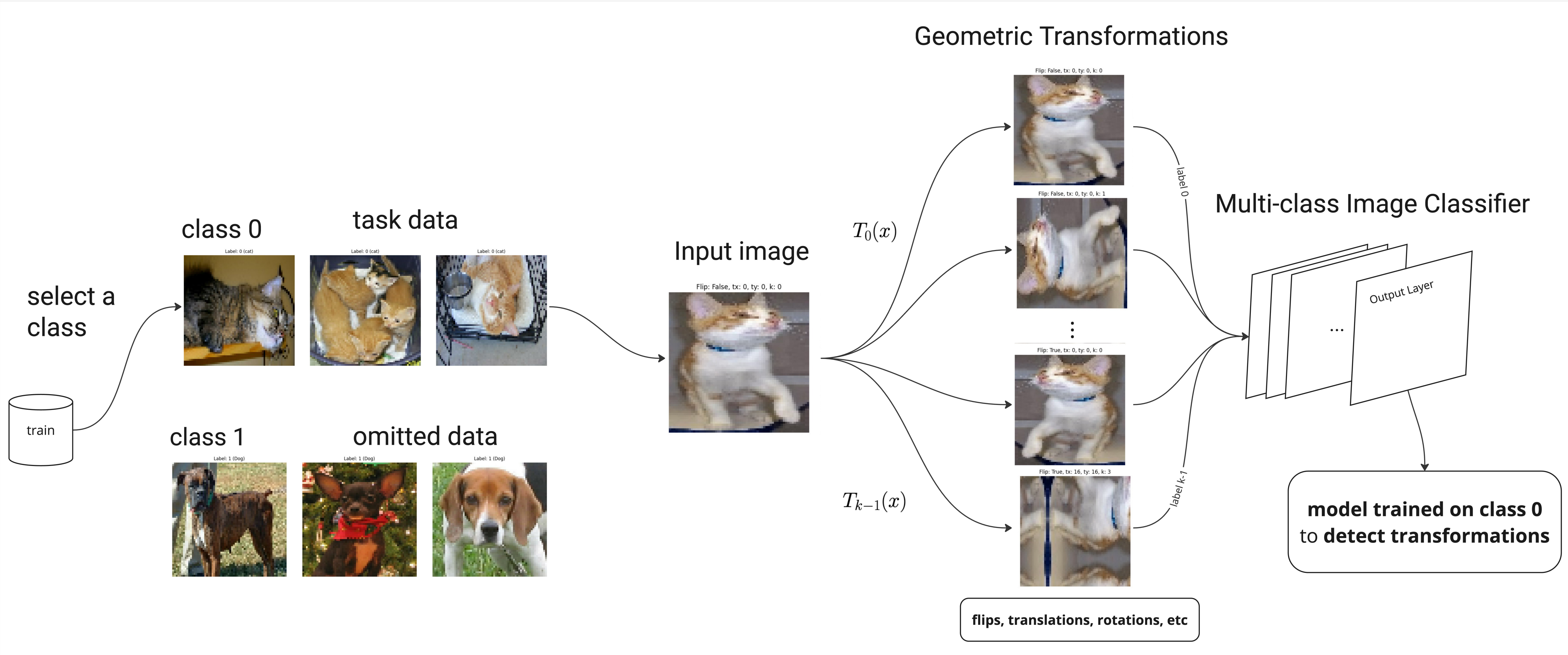


Figure: Illustration of the training structure on (Golan and El-Yaniv 2018).

Original Framework - Inference

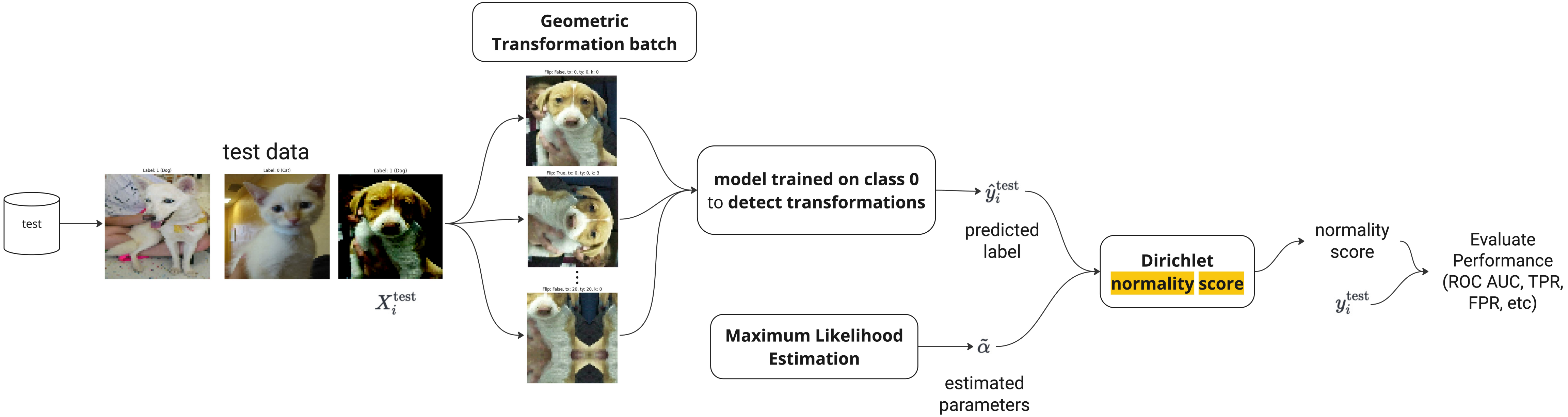


Figure: Illustration of the inference structure on (Golan and El-Yaniv 2018).

Model

(Golan and El-Yaniv 2018) utilizes a wide residual network (Zagoruyko 2016), involving:

- Around 53 layers,
- Convolutional Layers - spatial filtering,
- Activation layers - nonlinearity,
- Pooling layers - resizing,
- Batch normalization.

Brainstorming: possible experiments and extensions

- Image transformations
  - Try new transformations
- Normality score
  - Try new scores, with higher performance or computationally faster
- Uncertainty analysis

- Image transformations
  - Sensitivity analysis
  - Weight better transformations
- Hybrid models
  - Reconstruction (autoencoder) + Classification

1. Introduction
2. **Image Transformations**
3. Normality Score
4. Feature Localization
5. Uncertainty Estimation
6. Results and Conclusion

Framework Overview: Self-Labeling

Given a set of transformations $\mathcal{T} = \{T_0, \ldots, T_{k-1}\}$, where for each $1 < i < k-1$,

$$T_i : \mathcal{X} \rightarrow \mathcal{X} \tag{1}$$

and $T_0(x) = x$ is the identity transformation.
The self labeled set $S_{\mathcal{T}}$ is defined by:

$$S_{\mathcal{T}} := \{(T_j(x), j) : x \in S, T_j \in \mathcal{T}\} \tag{2}$$

So for any image $x \in S$, the label of the transformed image $T_j(x)$ is j.

Expanding the transformation set

Additional transformations were included:

- Zooming
- Random Crop
- Color jitter - random changes (brightness, contrast and saturation)
- Histogram equalization (

$$T = \left\{ T_{\text{old}} \circ T_s^{\text{zoom}} \circ T_b^{\text{crop}} \circ T_b^{\text{jitter}} \circ T_b^{\text{hist eq}} : b \in \{T, F\}, s \in \{1.0, 1.3\}, \right\}$$

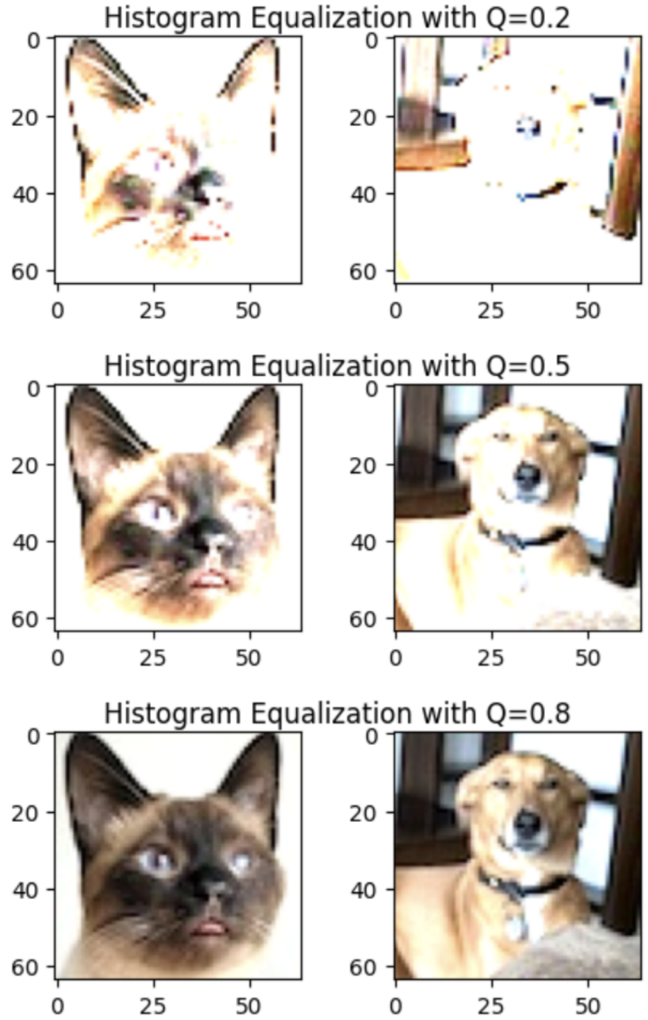
$$|T_{old}| = \underbrace{2}_{\substack{\text{flip} \\ \text{Y/N}}} \cdot \underbrace{3}_{\substack{\text{tx} \\ (0,-m,m)}} \cdot \underbrace{3}_{\substack{\text{ty} \\ (0,-m,m)}} \cdot \underbrace{4}_{\substack{\text{rotate} \\ (0,1,2,3)}} = 72$$

Figure: Example with additional transformations (2nd row).

Quantile Histogram Equalization

By adding a flexibility parameter Q, the histogram equalization normalized cdf was interpolated to the range $[0, Q]$, possibly minimizing the effects of equalization.

A default value of $Q = 0.7$ was fixed.



1. Introduction

2. Image Transformations

3. Normality Score

4. Feature Localization

5. Uncertainty Estimation

6. Results and Conclusion

Dirichlet normality score

Given a set of transformations $\mathcal{T} = \{T_0, \ldots, T_{k-1}\}$, and assuming a k-class model f_θ trained on a self-labeled set $S_\mathcal{T}$. Let $y(x) := \text{softmax}(f_\theta(x))$.

Each conditional distribution is approximated by $y(T_i(x))|T_i \sim \text{Dir}(\alpha_i)$, $\alpha_i \in \mathbb{R}_+^k$, $x \sim p_X(x)$, $i \sim \text{Uni}(0, k-1)$, and $p_X(x)$ is the real data probability distribution of "normal" samples.

The normality score of an image x is then:

$$n_S(x) = \sum_{i=0}^{k-1} (\tilde{\alpha}_i - 1) \cdot \log y(T_i(x))_j \qquad (3)$$

Normality score: new approach

(Taha and Hadi 2019) (Bereziński, Jasiul, and Szpyrka 2015)

The previous normality score relied on a Dirichlet score, which requires a maximum likelihood estimation (MLE) of parameters $\tilde{\alpha}_i$.

A **new approach is proposed, without the need of MLE** of parameters, via an **entropy score** H, as follows.

$$H(p) = -\sum_{i=1}^{N} p_i \log(p_i) \tag{4}$$

- Computationally cheaper (4.8x faster).

1. Introduction

2. Image Transformations

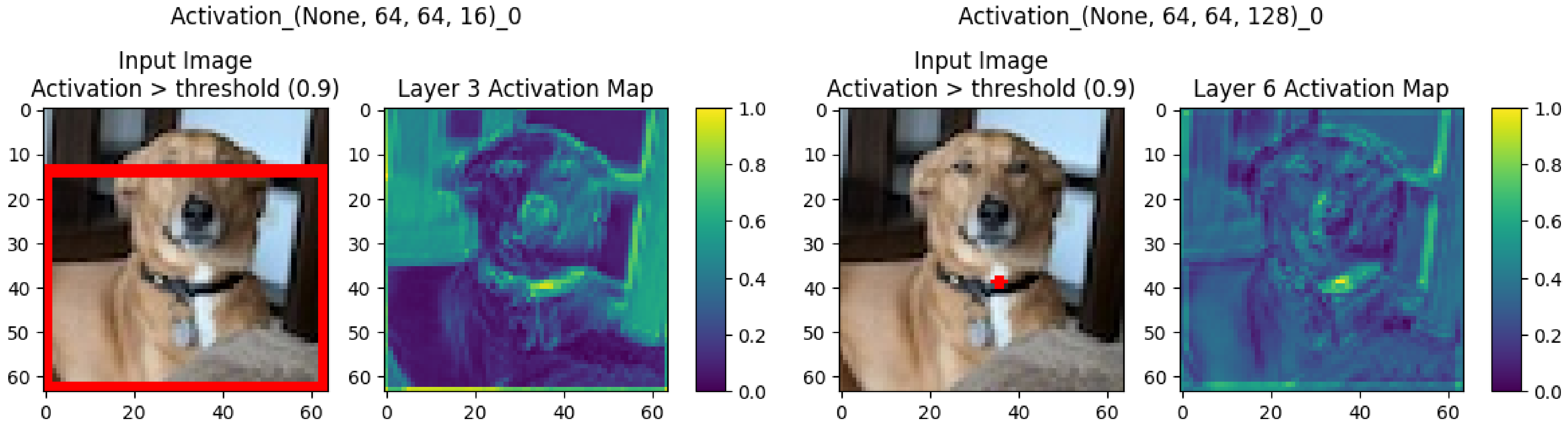
3. Normality Score

4. Feature Localization

5. Uncertainty Estimation

6. Results and Conclusion

Analyzing layers activations - Early features



Search for most salient features, one can notice high activation related to brightness, or the leash, on early features.

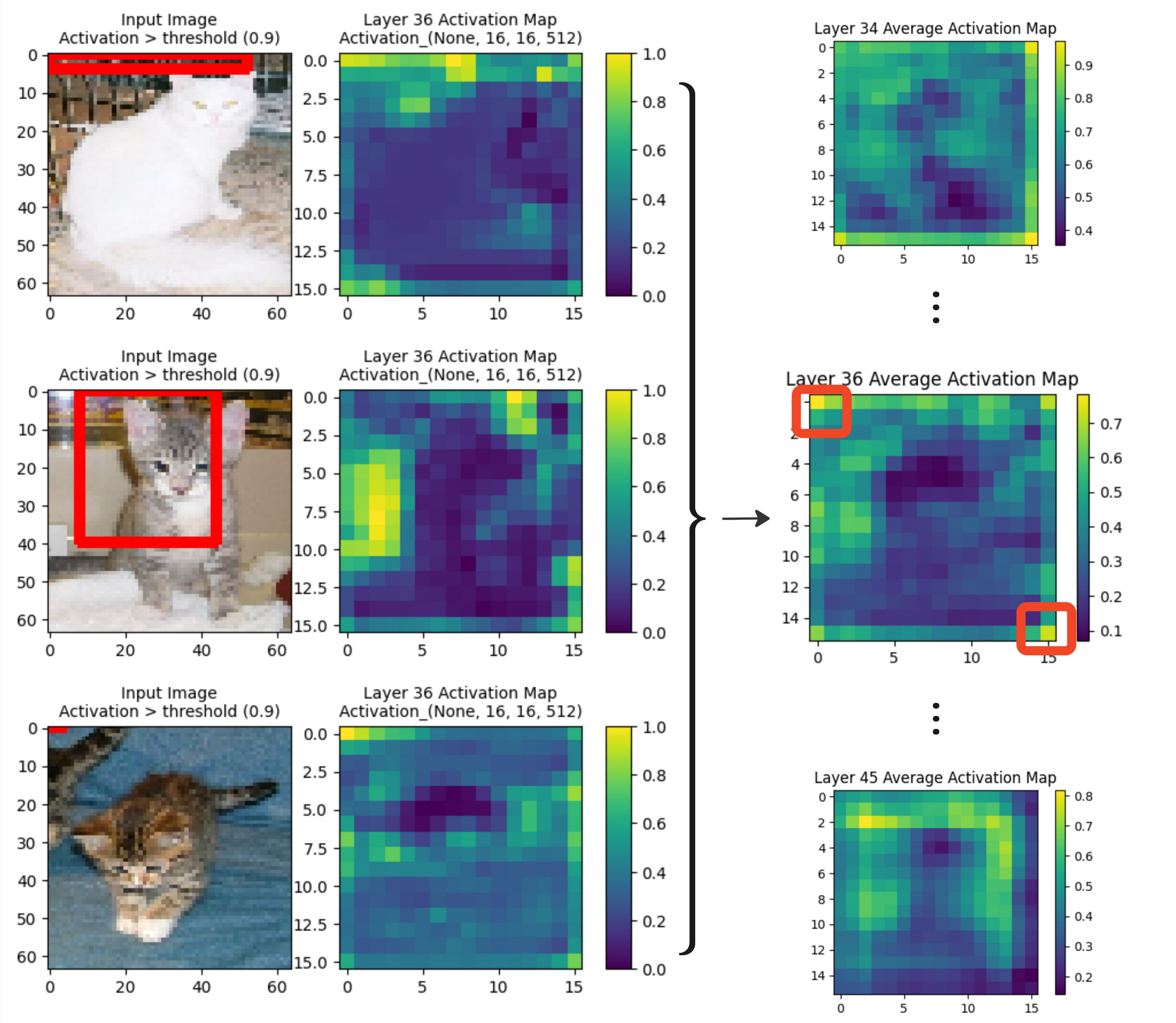
Average Convolutional Layer Activation

Aiming to recognize general patterns on a subset of N images $\{x_i\}_{i=1,\dots,N}$, the average activation map was extracted, for convolutional and activation layers, yielding $\overline{A}_k = \frac{1}{N}\sum_{i=1}^{N} A_k(x_i)$.

By setting a threshold τ, one can construct a mask of regions of higher importance.

$$\text{High Imp}(k) = \{\overline{A}_k(i,j) : \overline{A}_k(i,j) \geq \tau\} \quad (5)$$

The results show that boarders and corners had large importance, as well as the contour of the center.



Layer Activation analysis

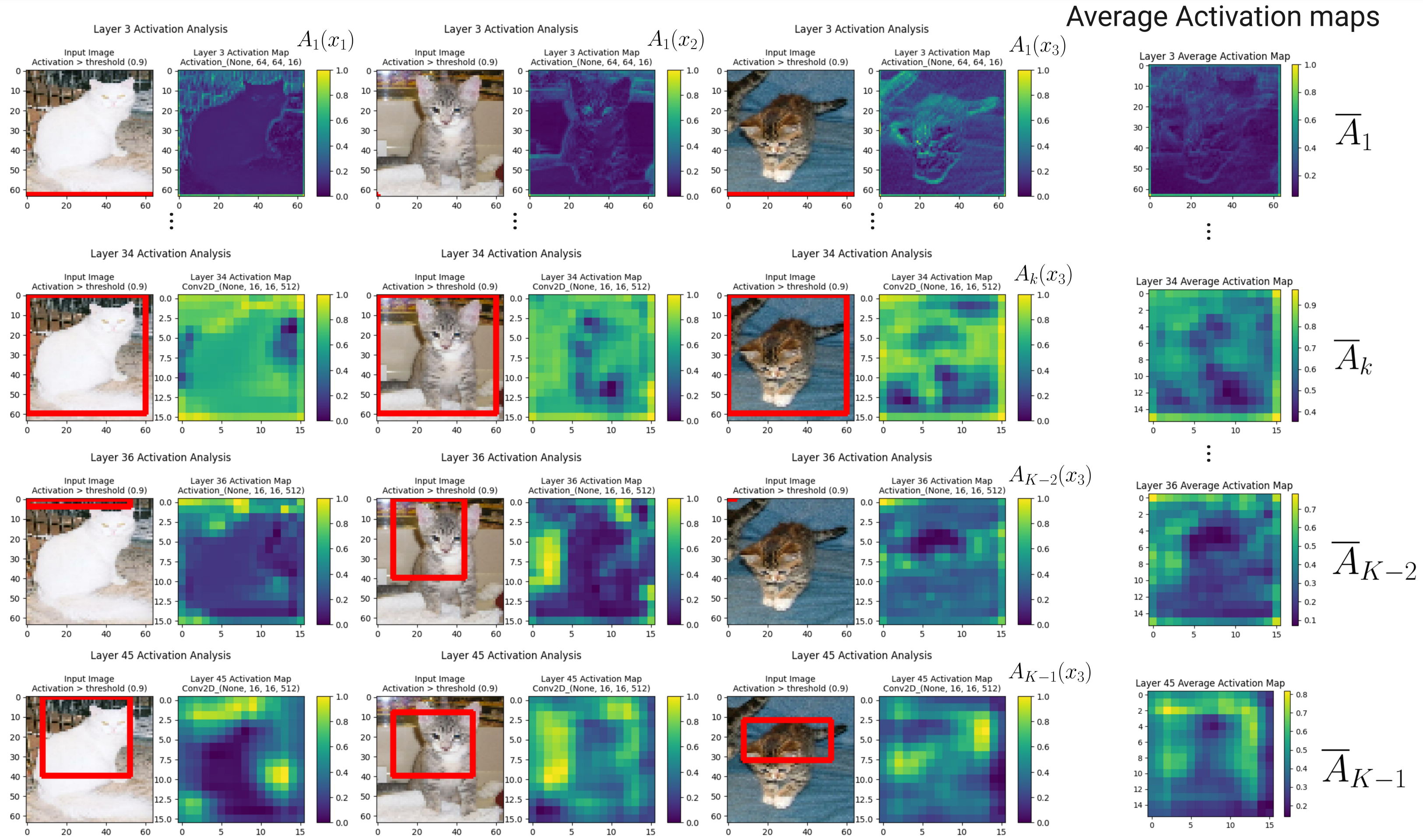


Figure: Illustration of the average activation map scheme.

Grad-CAM: Visual model explainability

By **weighting 2D-activations with the average gradient**, the region of largest importance is highlighted (Selvaraju et al. 2020).

Let $A^k \in \mathbb{R}^{H \times W}$ be the activation map for the k-th final convolutional layer of the CNN, and y^c be the score for class c. The gradient $\frac{\partial y^c}{\partial A^k_{i,j}}$ measures **importance of spatial locations** (i, j).

A **global importance weight** α^c_k representing how much the filter k contributes to class c is

$$\alpha^c_k = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \frac{\partial y^c}{\partial A^k_{i,j}} \tag{6}$$

The feature maps A^k are combined with weights α^c_k, constructing the heatmap for class c:

$$L^c_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k \alpha^c_k A^k\right) \tag{7}$$

Grad-CAM: Results

Figure: Grad-CAM examples: original image and rotated image.

1. Introduction

2. Image Transformations

3. Normality Score

4. Feature Localization

5. Uncertainty Estimation

6. Results and Conclusion

Uncertainty estimation: Monte Carlo Dropout

(Gal and Ghahramani 2016)

Given input x and a NN $f(x;\theta)$, MC dropout combines the **dropout regularization** and a **monte carlo sampling**, estimating a distribution of predictions $p(y|x;\theta)$ over labels y.

$$\hat{p}(y|x) \approx \frac{1}{N}\sum_{i=1}^{N}\hat{y}_i, \quad \hat{y}_i = f_D(x;\tilde{\theta}_i), \quad \tilde{\theta}_t \sim \text{Dropout}(\theta) \tag{8}$$

The **predictive uncertainty** is then $\text{Var}[\hat{y}] = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - \mathbb{E}(\hat{y}))^2$.
Goal: estimate the uncertainty σ_t^2 of a transformation prediction. High uncertainty and low confidence in the correct transformation indicate anomalous behaviour.

Uncertainty estimation example

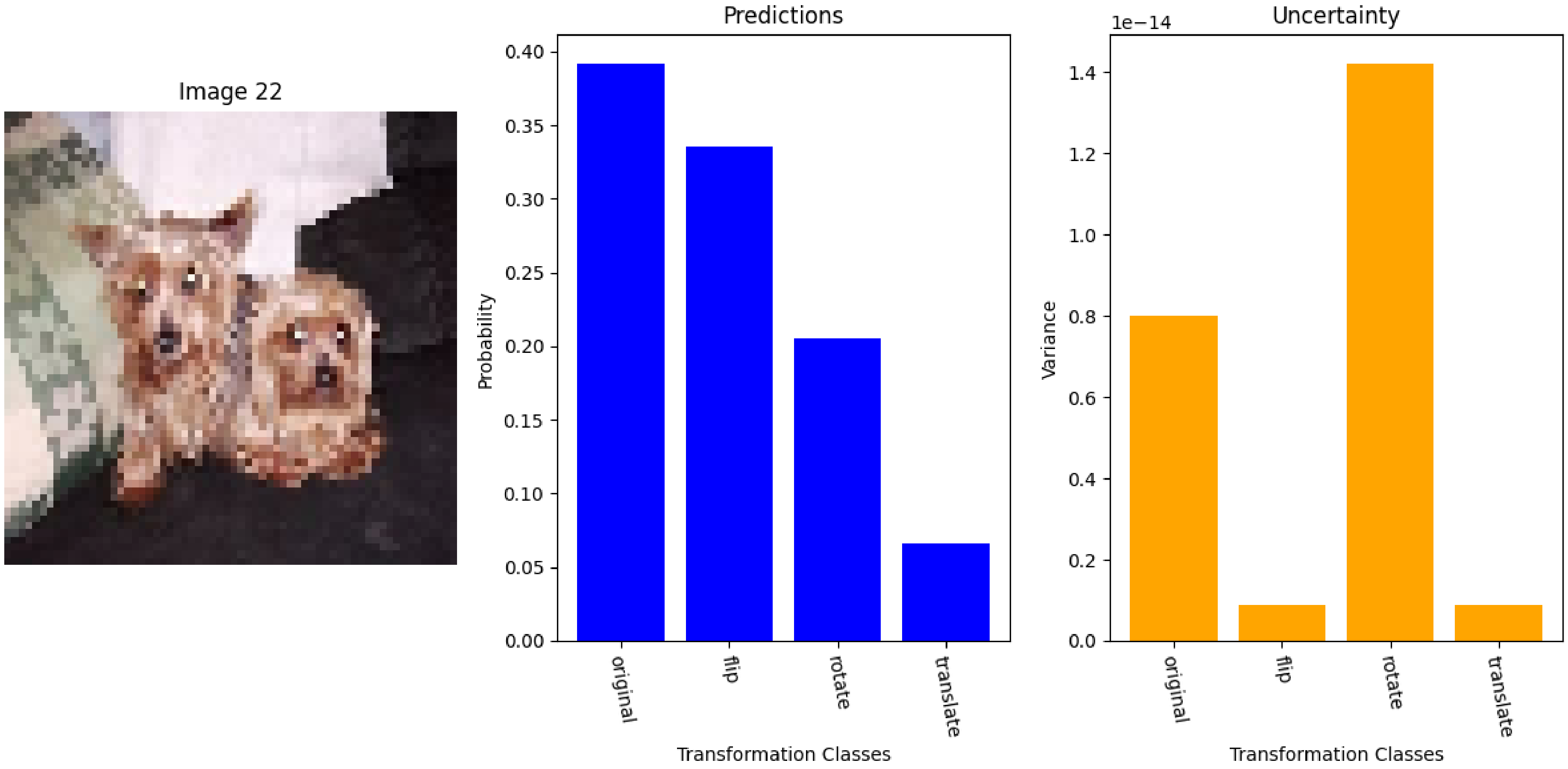


Figure: Example of model predictions and MC Dropout uncertainty estimation, with 10 epochs, 80 original training instances, and 50 MC passes.

1. Introduction
2. Image Transformations
3. Normality Score
4. Feature Localization
5. Uncertainty Estimation
6. Results and Conclusion

Results: alternative transformations and score ("mini" experiment)

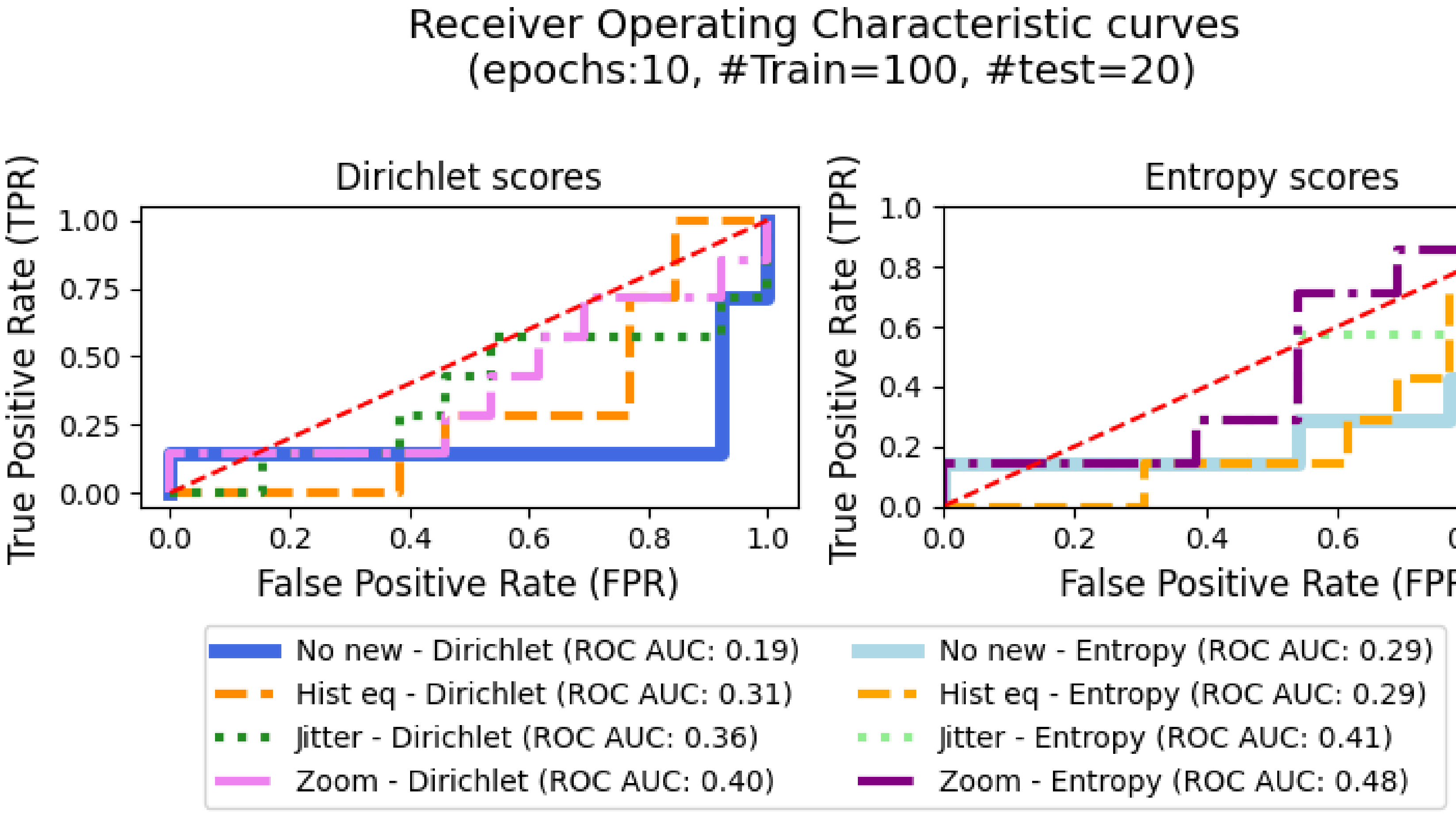


Figure: Small-scale experiment.

Results: alternative transformations and score (larger experiment)

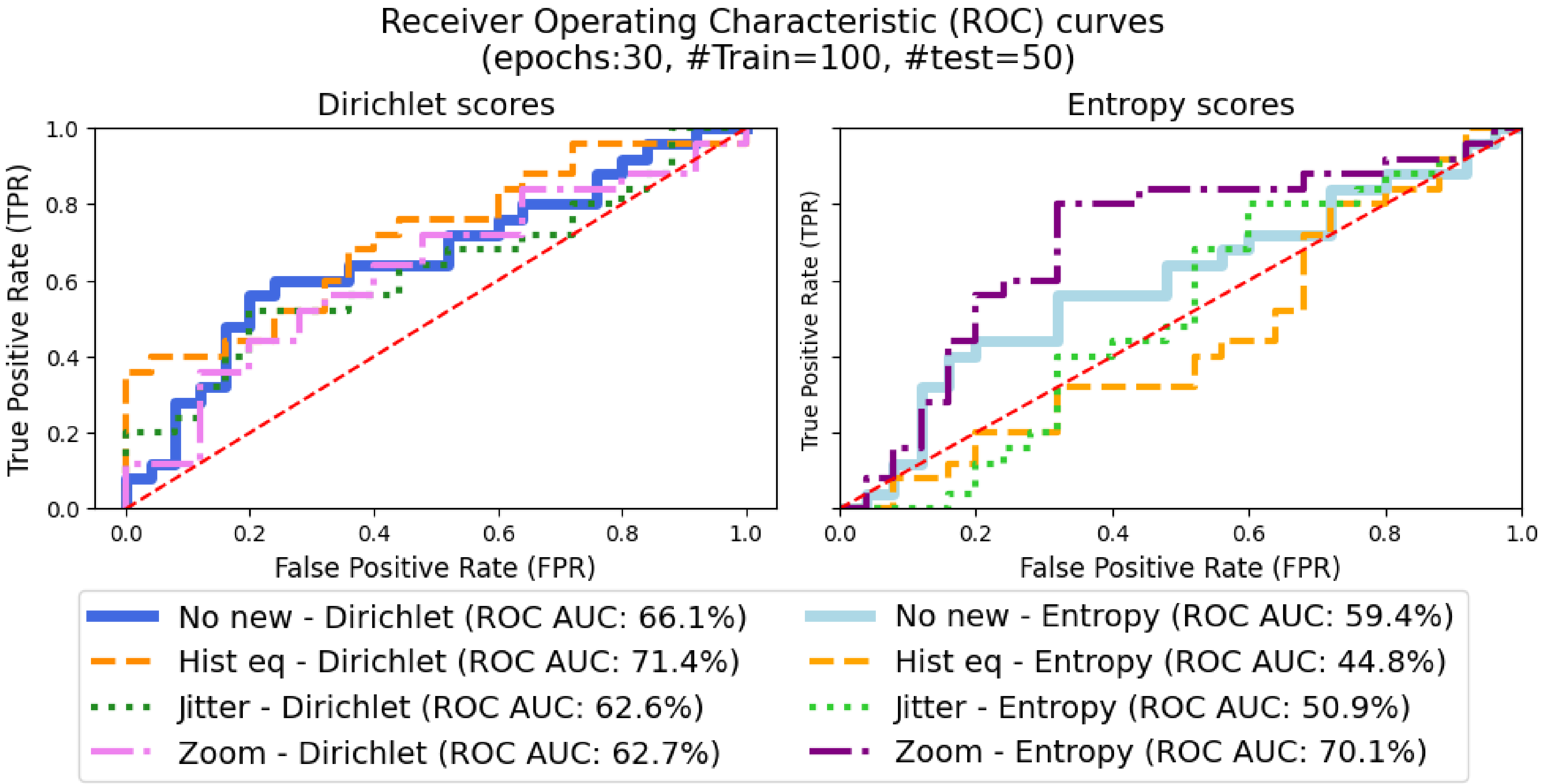


Figure: Results increasing the experiment scale.

Conclusion

- **Potential improvements:**
  - **New transformations**: especially **Zoom with Entropy** score and **Quantile Histogram Equalization with Dirichlet** score.
  - Shannon Entropy score.
- **Image borders and corners** showed **high relevance** for the geometric transformation detection model.
- **Uncertainty estimation** introduced an additional layer for ensuring model confidence.

- **Limitations, and further work**:
  - **Larger experiments**: Training on larger samples, and with more Epochs,
  - **More Monte Carlo steps** for the uncertainty analysis,
  - Testing on **different datasets**,
  - **Hybrid approaches** (reconstruction-based).

References I

Bereziński, P., B. Jasiul, and M. Szpyrka (2015). "An entropy-based network anomaly detection method". In: *Entropy* 17.4, pp. 2367–2408.

Gal, Y. and Z. Ghahramani (2016). "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In: *international conference on machine learning*. PMLR, pp. 1050–1059.

Golan, I. and R. El-Yaniv (2018). "Deep Anomaly Detection Using Geometric Transformations". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc. URL: `https://proceedings.neurips.cc/paper_files/paper/2018/file/5e62d03aec0d17facfc5355dd90d441c-Paper.pdf`.

Selvaraju, R. R., M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra (2020). "Grad-CAM: visual explanations from deep networks via gradient-based localization". In: *International journal of computer vision* 128, pp. 336–359.

Taha, A. and A. S. Hadi (2019). "Anomaly detection methods for categorical data: A review". In: *ACM Computing Surveys (CSUR)* 52.2, pp. 1–35.

Zagoruyko, S. (2016). "Wide residual networks". In: *arXiv preprint arXiv:1605.07146*.