Karlsruher Institut für Technologie

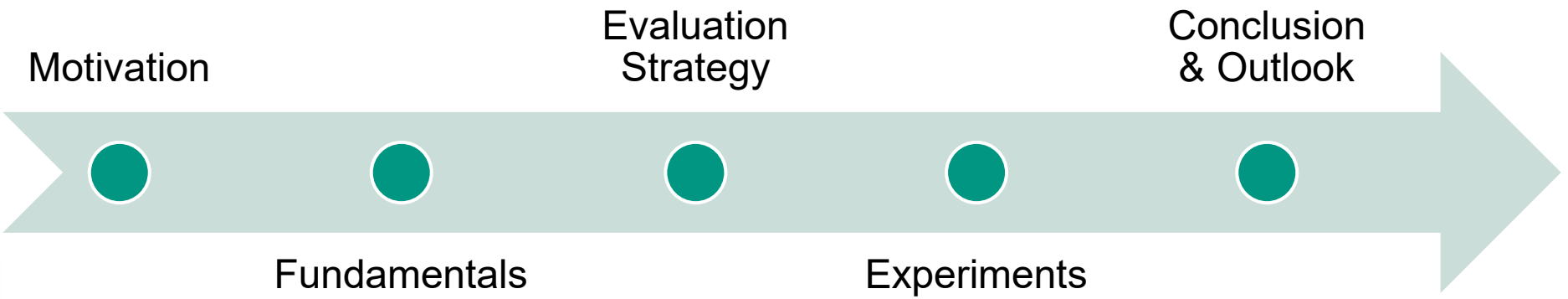
Semantic Segmentation and Uncertainty Estimation with Vision Transformers in an Industrial Context

Institute of Photogrammetry and Remote Sensing, Department of Civil Engineering, Geo and Environmental Sciences

Lili Gao

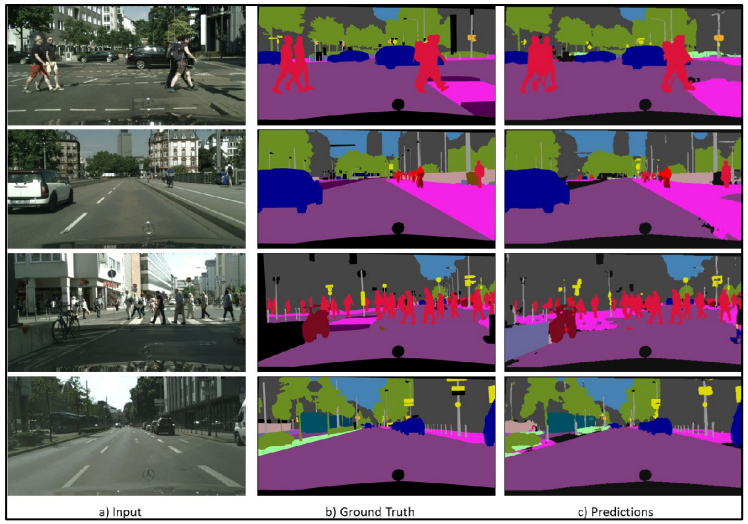
Reviewer: Prof. Dr.-Ing. Markus Ulrich
Advisor: Steven Landgraf und Kira Wursthorn

Agenda

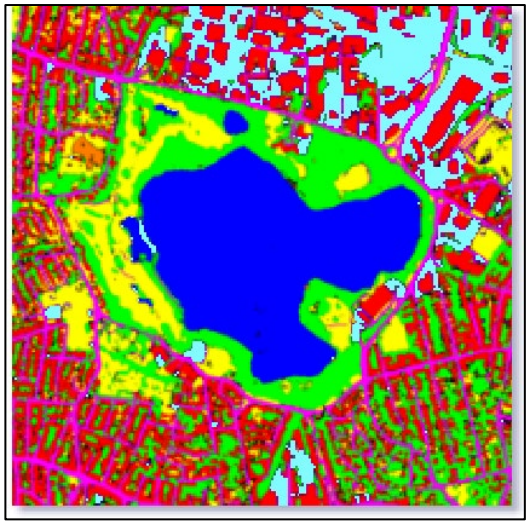

Motivation
Fundamentals
• Semantic Segmentation
• Vision Transformer
• Uncertainty Estimation
Evaluation Strategy
Experiments & Results
Conclusion & Outlook

MOTIVATION

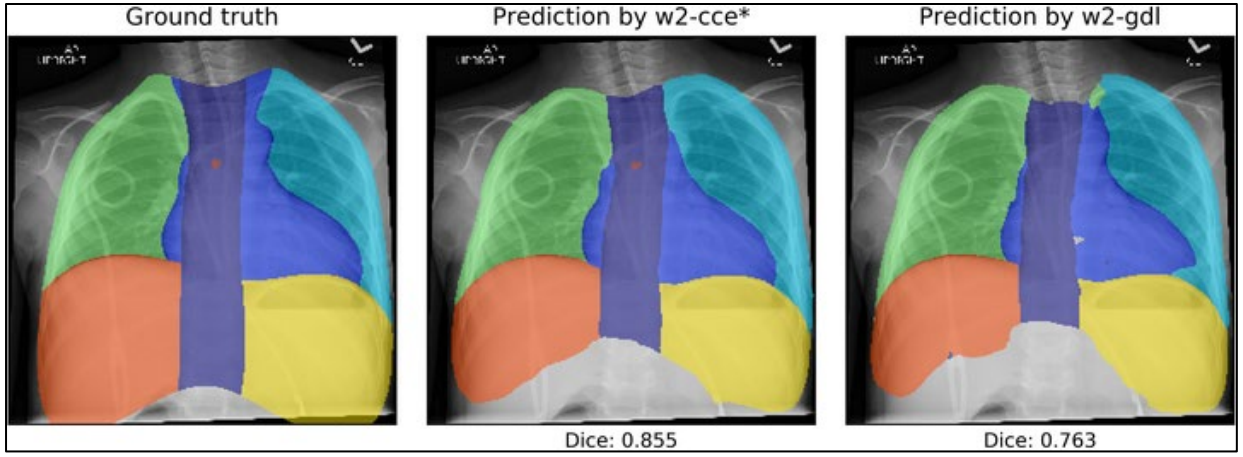
Motivation: Semantic Segmentation



Y.G. et al., 2018



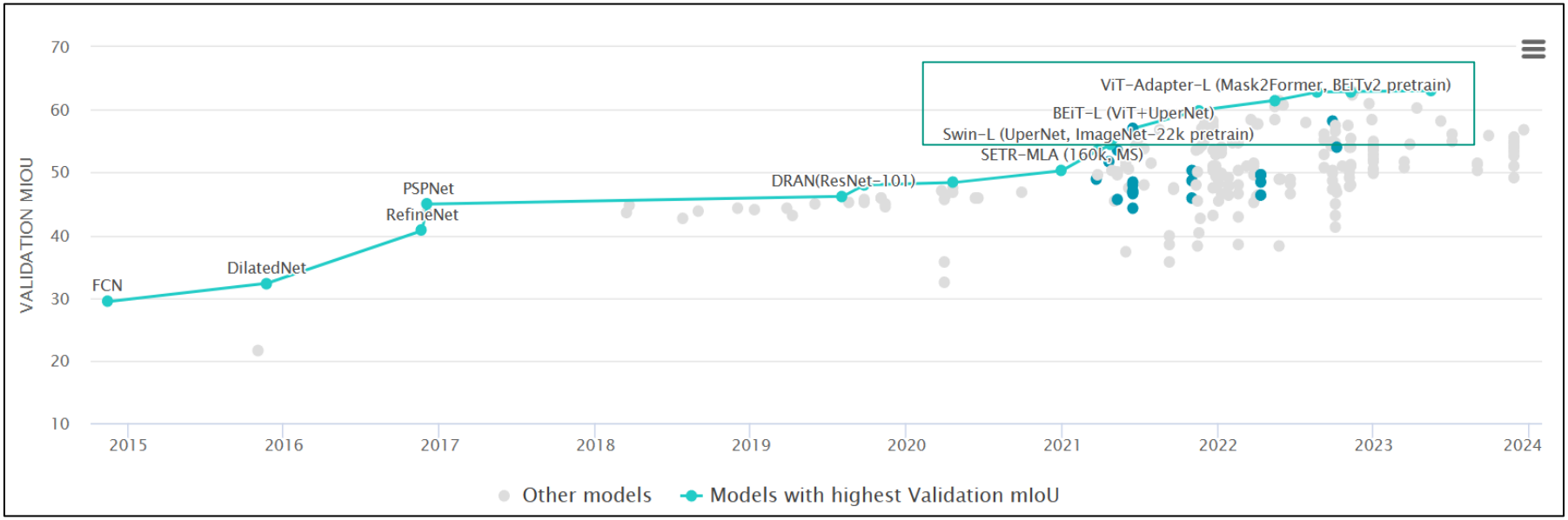
Carl, 2018



Holste et al., 2020

Motivation: ViT

Semantic Segmentation on ADE20K

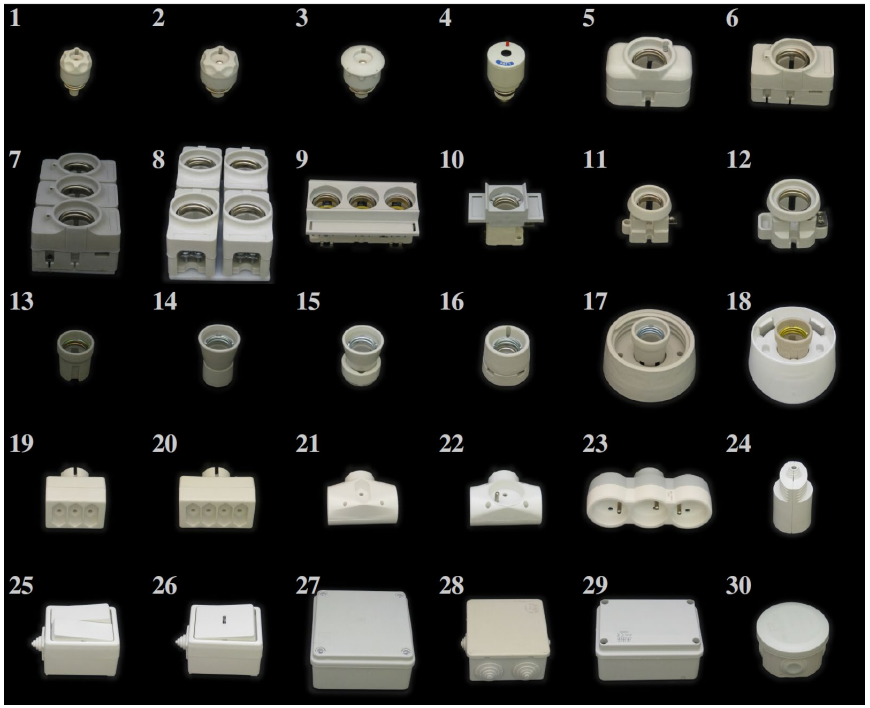


https://paperswithcode.com/sota/semantic-segmentation-on-ade20k

→ How is the performance of ViT in segmenting industrial objects?

Motivation: T-LESS Dataset

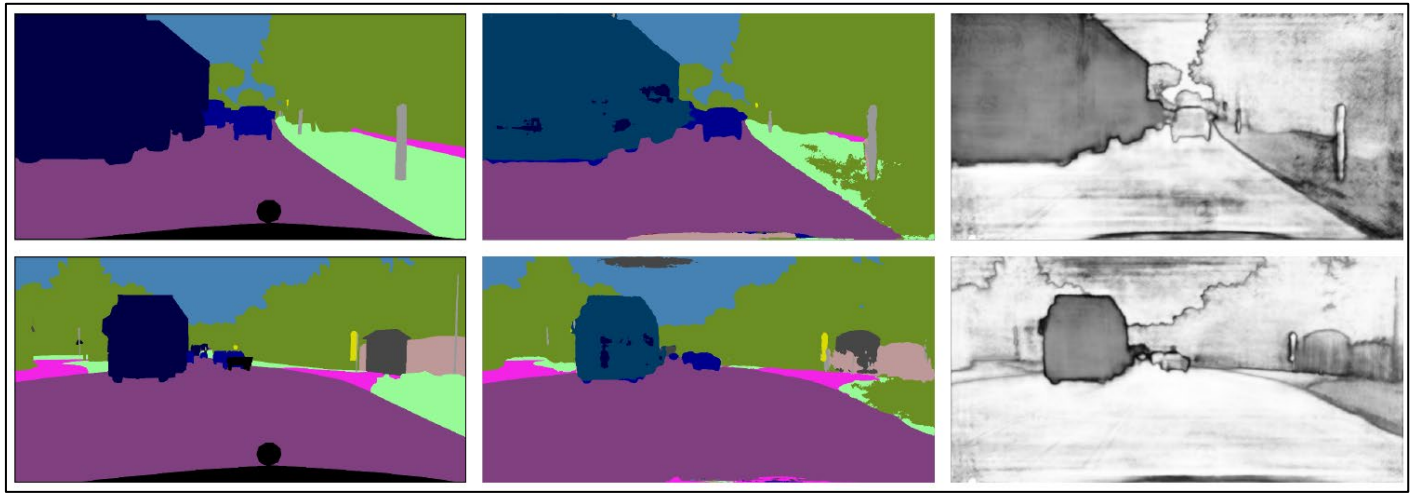
- BOP Challenge (bop.felk.cvut.cz): Benchmark for 6D Object Pose Estimation
- 30 industry-relevant objects: texture-less & colorless
- Training images:
  - 38K from sensor data captured by Primesense CARMINE 1.09 - individual objects
  - 50K synthetic data generated by BlenderProc - 50 scenes with varying complexity
- Testing images: 1K sensor-captured data - 20 scenes with varying complexity



T. Hodaň et al., 2017

Motivation: Uncertainty Estimation

- Better interpret the reliability of prediction
- Enable the special handling of uncertain prediction
- Identify the potential inaccurate predictions
- Crucial in fields like autonomous driving, medical diagnostics
- In industrial context
  - enhance reliability in automated systems
  - ensure more efficient operations



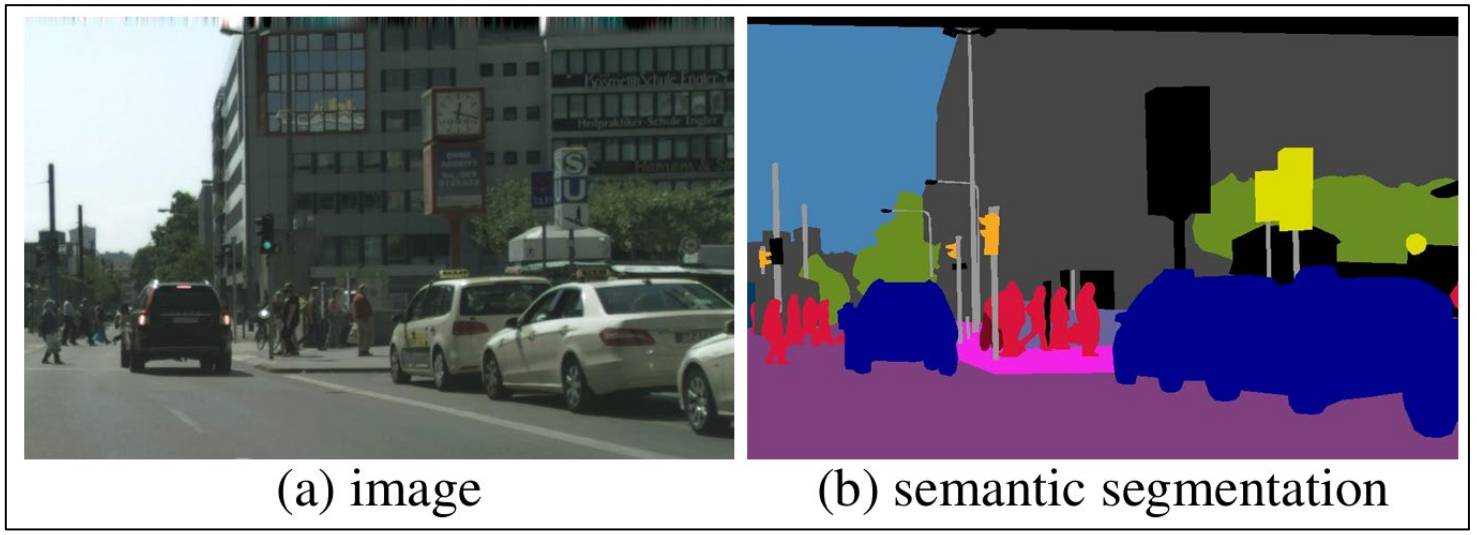
Ground truth labels | Segmentation Predictions | Uncertainty Predictions

Mukhoti et al., 2017

THEORETICAL FUNDAMENTALS

Semantic Segmentation

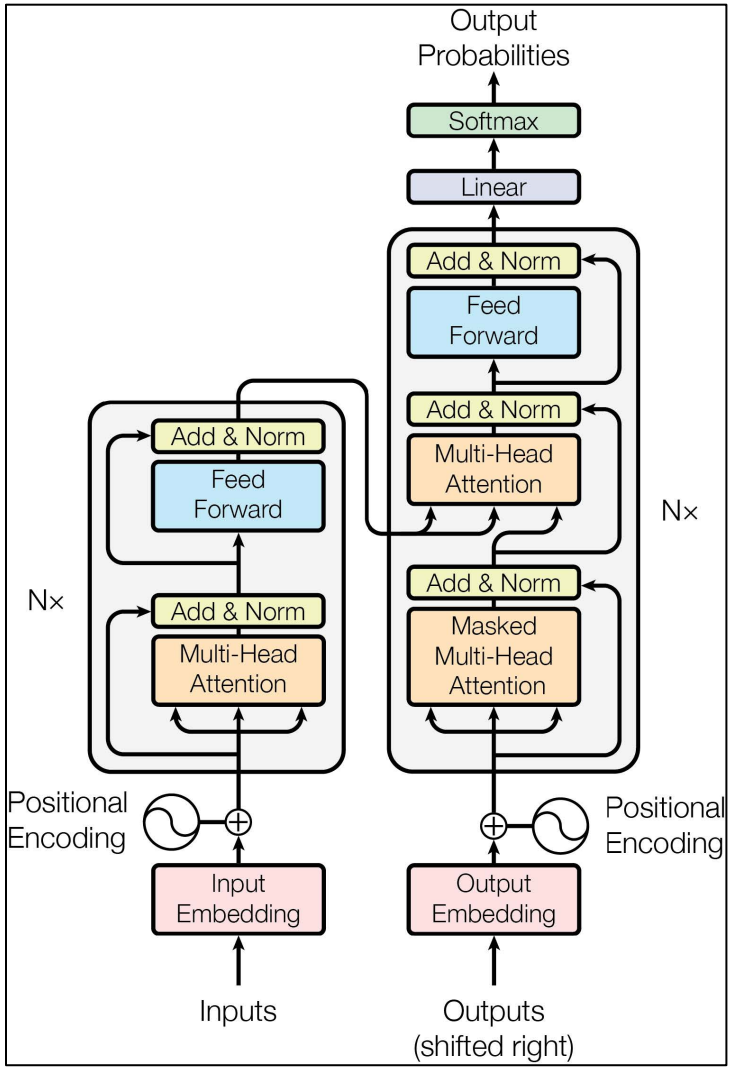
- A fundamental computer vision task
- Assign a class label to every single pixel of an input image
- Ground truth / Output: segmentation maps



Kirillov et al., 2019

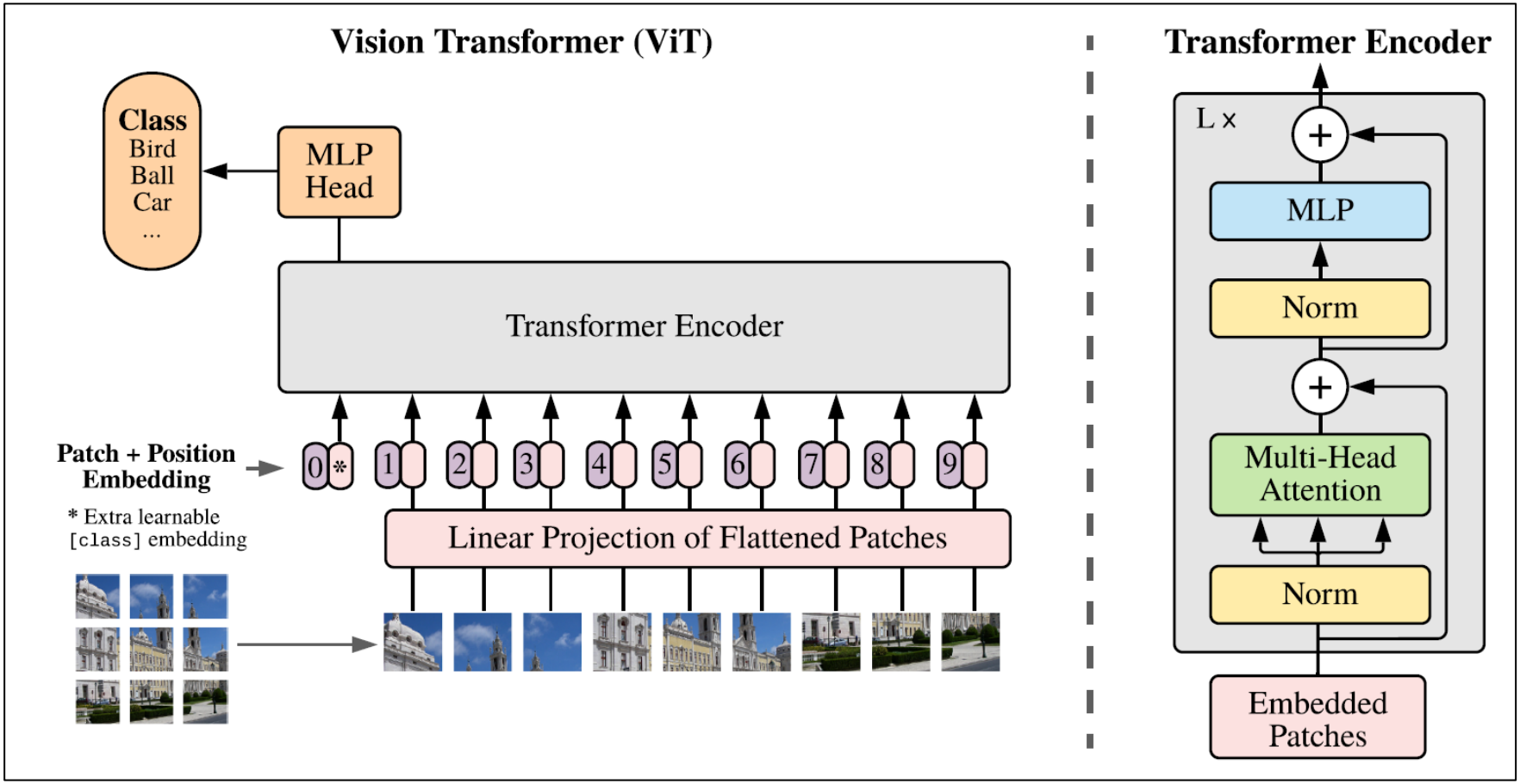
Transformer

- Employs self-attention mechanisms to process input sequences in parrallel
- Positional Encoding: enables the consideration of sequence order
- Encoder-Decoder Structure:
  - Encoder processes the input sequence
  - Decoder generates the output sequence



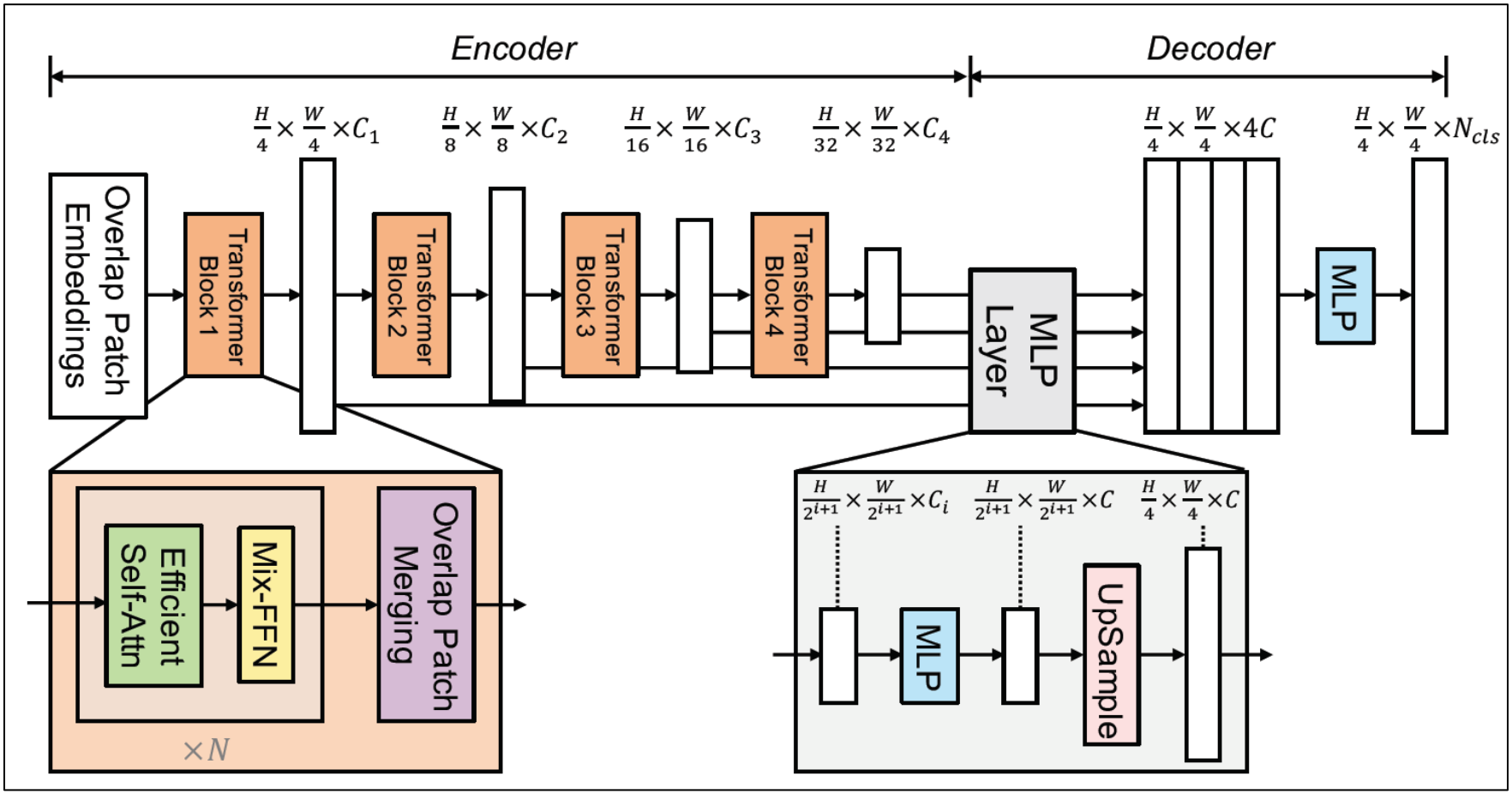
Vaswani et al., 2017

Vision Transformer



Dosovitskiy et al., 2021

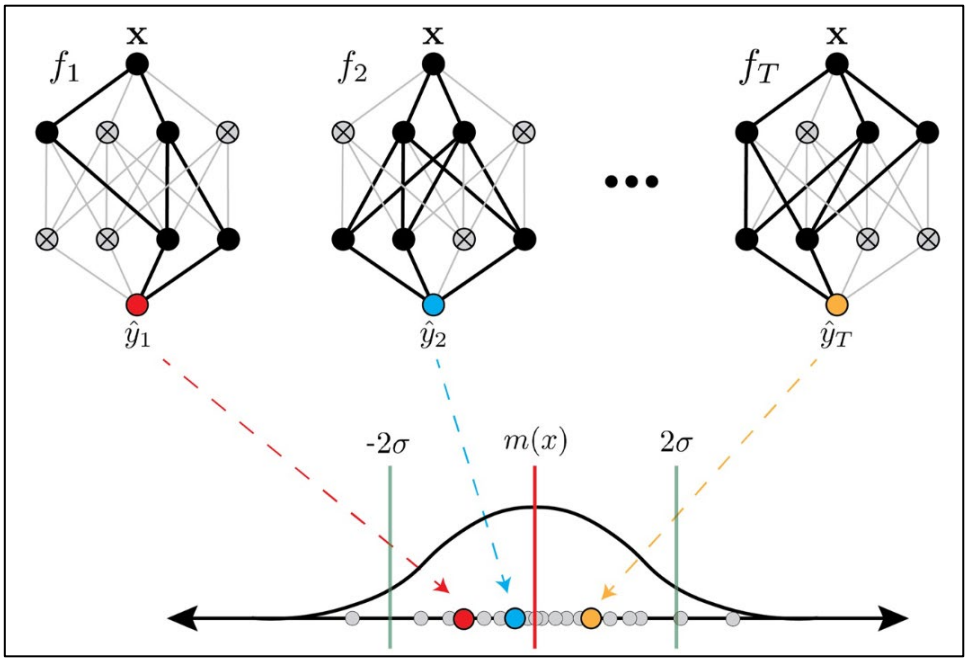
Selected model: SegFormer



Xie et al., 2021

Uncertainty Estimation: Monte Carlo Dropout

- Apply dropout not only in training but also during inference (test phase)
- Dropout layer = a Bernoulli distribution over the layer
- predictions with dropout after each hidden layer = Monte Carlo samples
- **mean** / median → prediction
- **standard deviation** / variance / entropy → uncertainty



Van et al., 2023

$$\text{Variance: } \sigma^2 = \frac{1}{N-1} \cdot \sum_{n=1}^{N} (x(n) - \mu)^2$$

$$\text{Standard Deviation: } \sigma = \sqrt{\sigma^2}$$

$$\text{Entropy: } en = -\sum_{i=0}^{x_{max}} (p(x_i) \cdot \log_e(p_{x_i}))$$

Gal et al., 2016

EVALUATION STRATEGY

General Approach

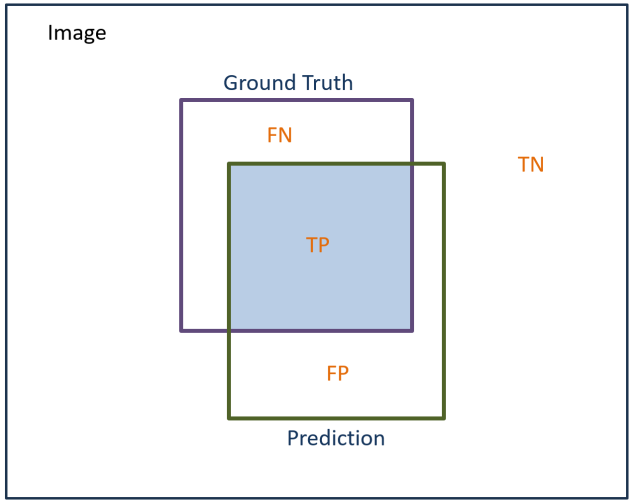
- Semantic segmentation
  - Investigation the influence of training dataset
  - Quality analysis of different model backbones
  - Investigation the influence of data augmentation
  - Investigation of the variation of the learning rates
- Uncertainty estimation
  - Investigation the influence of sample size
  - Investigation of different dropout rates
  - Investigation of different uncertainty metrics (standard deviation vs. entropy)
- Experiments conducted in 2 steps
  - Training phase
  - Testing phase

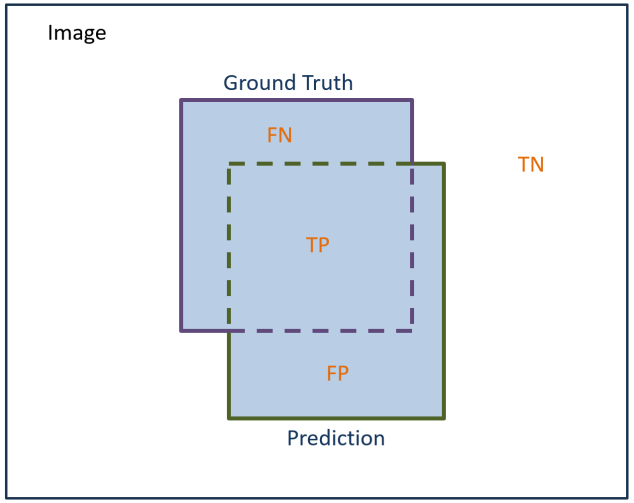
Mean Intersection-Over-Union (Mean IoU)

$$IoU_{class_i} = \frac{TP}{TP + FP + FN}$$

$$IoU_{class_i} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

$$Mean\ IoU = \frac{1}{n} \cdot \sum_{i=1}^{n} IoU_{class_i}$$





Expected Calibration Error (ECE)

- assess the calibration of estimated probabilities generated by the model
- Calibration: the consistency between the predicted probabilities and the actual frequency of events

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|$$

- With

$$\mathrm{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}(\hat{y}_i = y_i)$$

$$\mathrm{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i$$

- n : the total number of samples
- B_m: the m-th bin
- $\mathbf{1}$: indicator function
- $\hat{p}_i$: the maximum estimated probability

Uncertainty Quality Metrics

- Assumptions:
  - if a model is confident about its prediction, it should be accurate.
  - if a model is inaccurate on an output, it should be uncertain about it.
- uncertainty threshold: the average uncertainty of all pixels over the dataset
- Metrics:

$$p(\mathbf{accurate}|\mathbf{certain}) = \frac{n_{ac}}{n_{ac} + n_{ic}}$$

$$p(\mathbf{uncertain}|\mathbf{inaccurate}) = \frac{n_{iu}}{n_{ic} + n_{iu}}$$

$$\mathrm{PAvPU} = \frac{n_{ac} + n_{iu}}{n_{ac} + n_{au} + n_{ic} + n_{iu}}$$

Mukhoti et al., 2018

Implementation Details

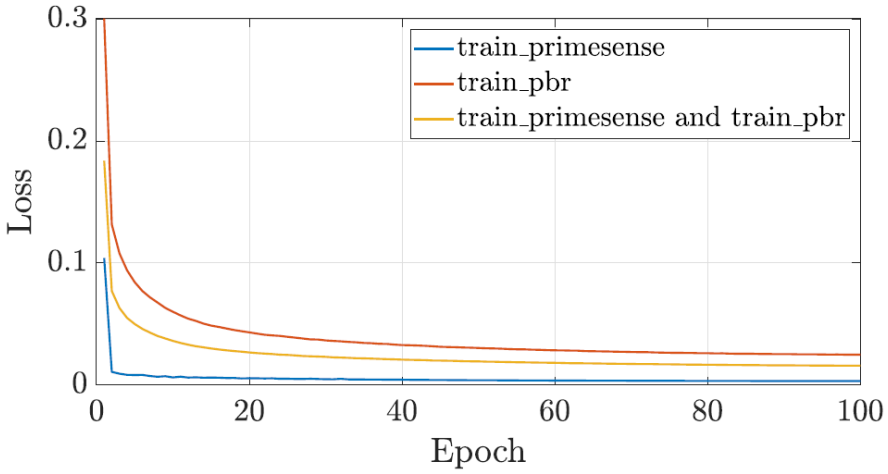
- Training epochs: 100
- 31 segmentation classes: 30 object classes + 1 rejection class
- Batch size: for train = 8, for validation = 4
- Loss function: cross-entropy
- Optimizer: AdamW
- Initial weight decay: 0.01
- Scheduler: polynomial learning rate scheduler with a default factor of 1.0
- Learning rate multiplier (the ratio of the learning rate between the encoder and decoder): 1
- Gradient clipping: global norm = 0.5 to prevent exploding gradients
- Training hardware: 4 GPUs with Distributed Data Parallel (DDP) strategy
- Testing metrics:
  - mean IoU: torchmetrics.JaccardIndex for multiclass tasks for 31 classes
  - ECE: torchmetrics.Calibrat with 10 bins

EXPERIMENTS AND RESULTS

Part 1: Semantic Segmentation
Part 2: Uncertainty Estimation

Part 1: Influence of Training Dataset

- **Dataset:**
  - **Synthetic dataset: *train_pbr***
  - **sensor-captured dataset: *train_primesense***
  - **Both datasets**
- Model backbone: SegFormer-B2
- Learning rate: 6×10^{-5}

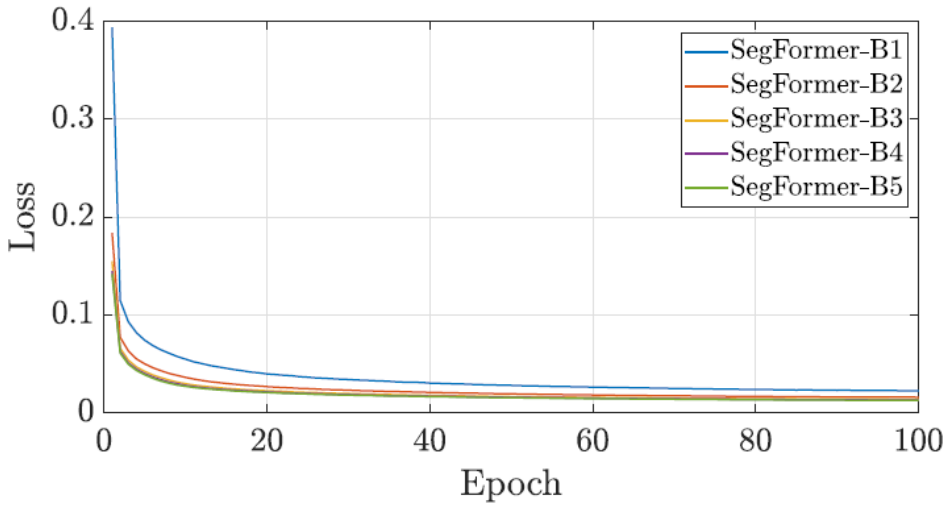


Train losses

Training Dataset	Mean IoU
train_primesense	4.33%
train_pbr	33.44%
both datasets	54.28%

Part 1: Influence of Model Backbones

- Both datasets
- **Model backbone: SegFormer-B1 to SegFormer-B5**
- Learning rate: 6×10^{-5}



Train losses

Model backbone	Mean IoU	ECE
SegFormer-B1	44.69%	14.30%
SegFormer-B2	54.27%	10.80%
SegFormer-B3	62.45%	15.61%
SegFormer-B4	63.03%	12.23%
SegFormer-B5	63.92%	8.92%

Part 1: Influence of Data Augmentation

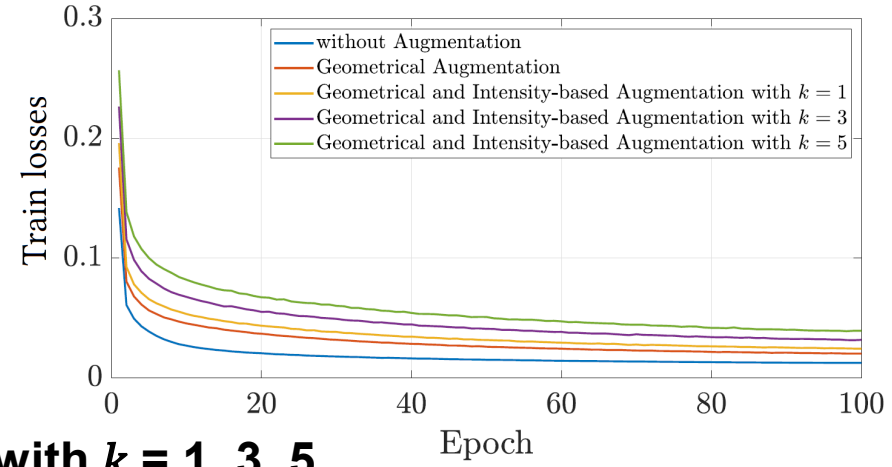
- Weak **geometrical** augmentation
  - Random Scale: Randomly scale the image with a ratio of 0.5-2.0.
  - Random Flip: Randomly flip the image horizontally with a probability of 0.5.
  - Random Crop: Randomly crop a region from the image to the size of 512 × 512.

- Random **intensity-based** augmentation: randomly selects k of
  - Identity: Return the original image.
  - Autocontrast: Maximize the contrast of an image.
  - Equalize: Equalize the histogram of an image.
  - Gaussian blur: Performs Gaussian blurring on the image.
  - Contrast: Adjust the contrast of the image by a factor chosen from [0.5, 1.5].
  - Sharpness: Adjust the sharpness of the image by a factor chosen from [0.5, 1.5].
  - Saturation: Adjust the saturation of the image by a factor chosen from [0.5, 1.5].
  - Brightness: Adjust the brightness of the image by a factor chosen from [0.5, 1.5].
  - Hue: Adjust the hue of the image by a factor chosen from [0.5, 1.5].
  - Posterize: Reduce each pixel to [4, 8] bits for each color channel.

Zhao et al., 2023

Part 1: Influence of Data Augmentation

- Both datasets
- Model backbone: SegFormer-B5
- Learning rate: 6×10^{-5}
- **Data augmentation:**
  - **None**
  - **Only geometrical**
  - **Geometrical and intensity-based with k = 1, 3, 5**

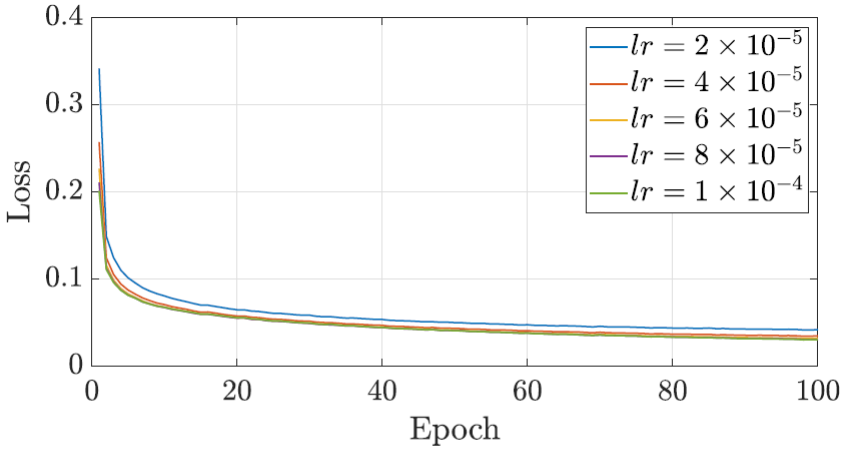


Augmentation	Mean IoU	ECE
Without data augmentation	63.92%	8.92%
With geometrical augmentation	74.81%	6.65%
With geometrical and intensity-based augmentation with $k = 1$	76.35%	6.07%
With geometrical and intensity-based augmentation with $k = 3$	79.40%	4.29%
With geometrical and intensity-based augmentation with $k = 5$	79.22%	5.30%

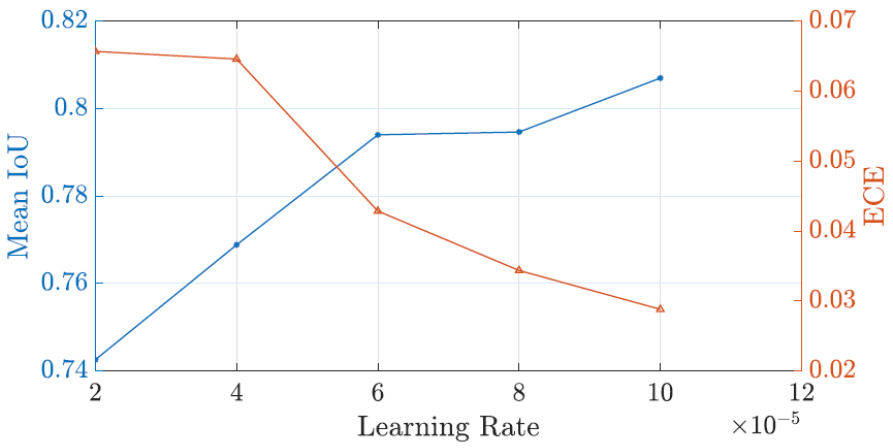
Part 1: Influence of Learning Rates

- Both datasets
- Model backbone: SegFormer-B5
- Data augmentation: Geometrical and intensity-based with k = 3
- **Learning rate: 2×10^{-5} to 2×10^{-4}**

Model backbone	Mean IoU	ECE
$lr = 2 \times 10^{-5}$	74.26%	6.57%
$lr = 4 \times 10^{-5}$	76.88%	6.46%
$lr = 6 \times 10^{-5}$	79.40%	4.29%
$lr = 8 \times 10^{-5}$	79.47%	3.44%
$lr = 1 \times 10^{-4}$	80.70%	2.88%

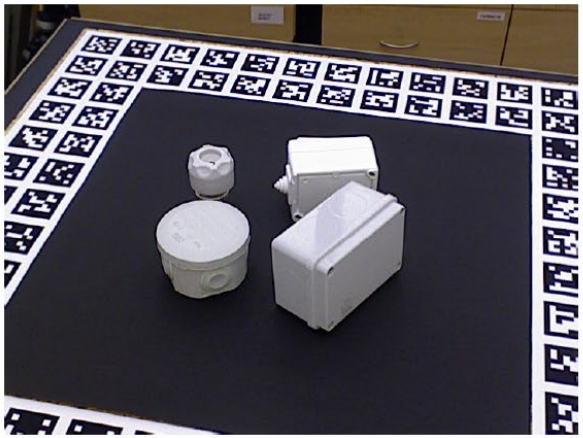


(a) Train losses.

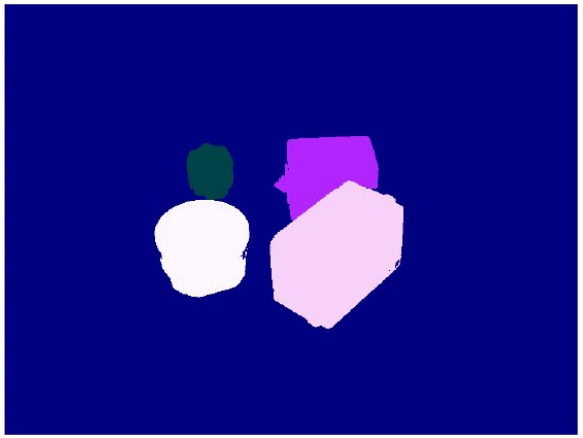


(b) Mean IoU and ECE on the testing dataset with respect to learning rates.

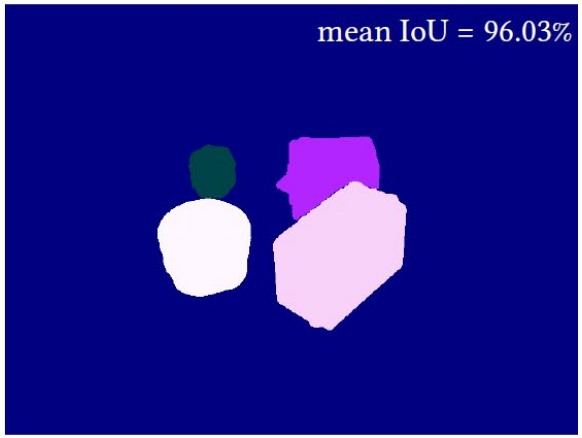
Part 1: Qualitative Results



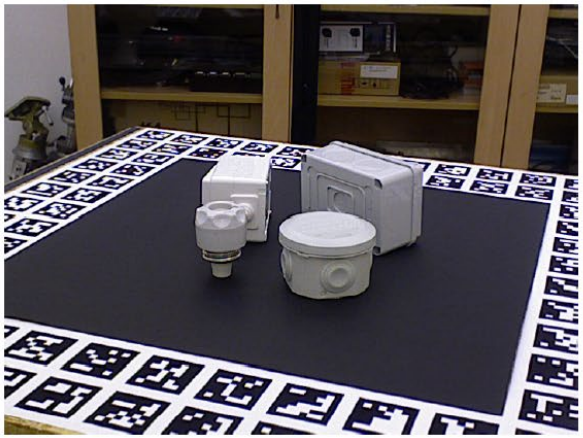
(a) Scene 1, image 236.



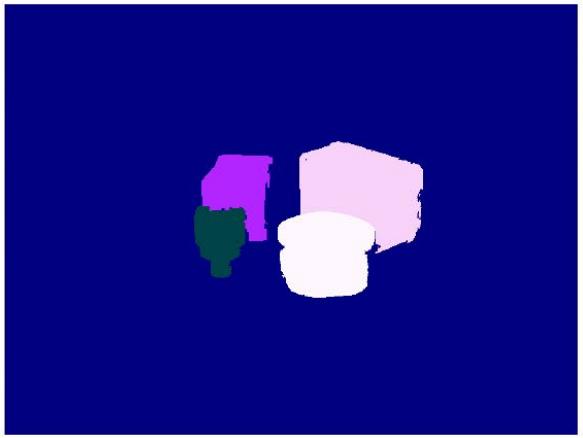
(b) Ground truth.



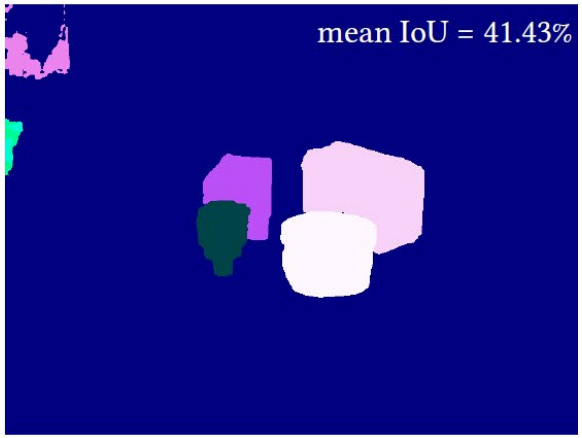
(c) Prediction.



(d) Scene 1, image 364.



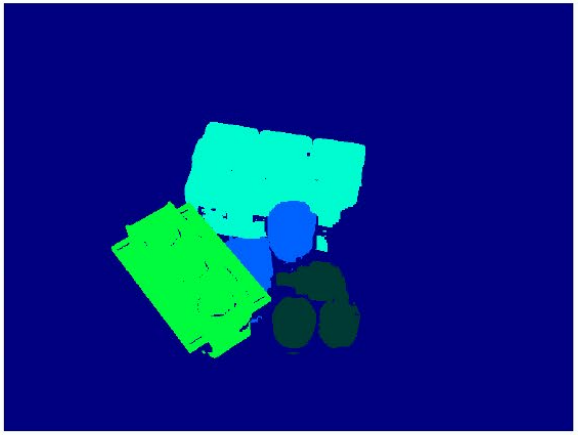
(e) Ground truth.



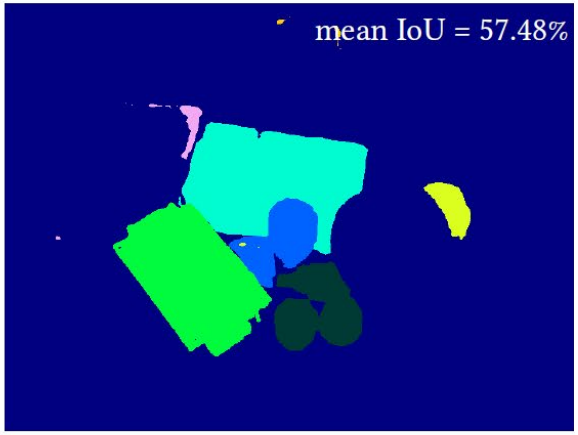
(f) Prediction.

Part 1: Qualitative Results

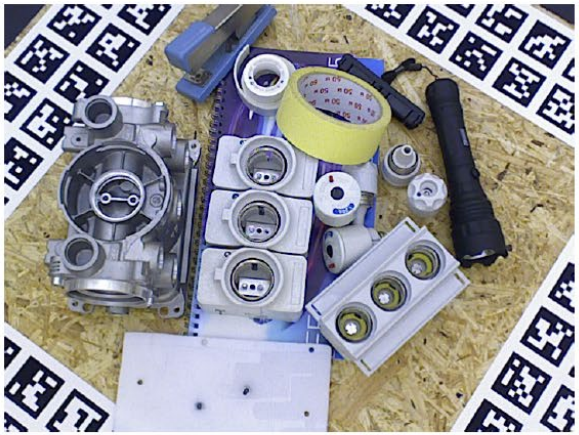
(a) Scene 18, image 206.



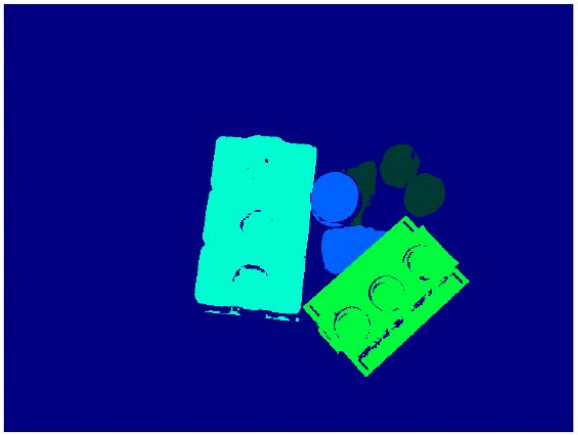
(b) Ground truth.



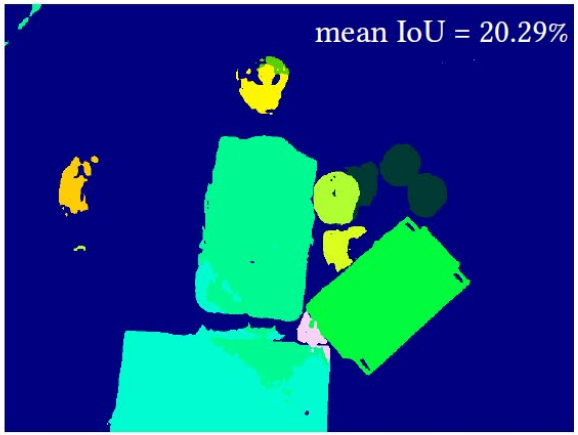
(c) Prediction.



(d) Scene 18, image 43.



(e) Ground truth.



(f) Prediction.

Part 2: Uncertainty Estimation

- MC Dropout method combined with SegFormer

- **Sample sizes** during inference: $2, 5, 10, 20$ & 200
- 4 Models trained with **dropout rates** $10\%, 20\%, 30\%$ & 50%, respectively
- **Uncertainty metrics**: standard deviation vs. entropy
- Evaluation with the uncertainty quality metrics

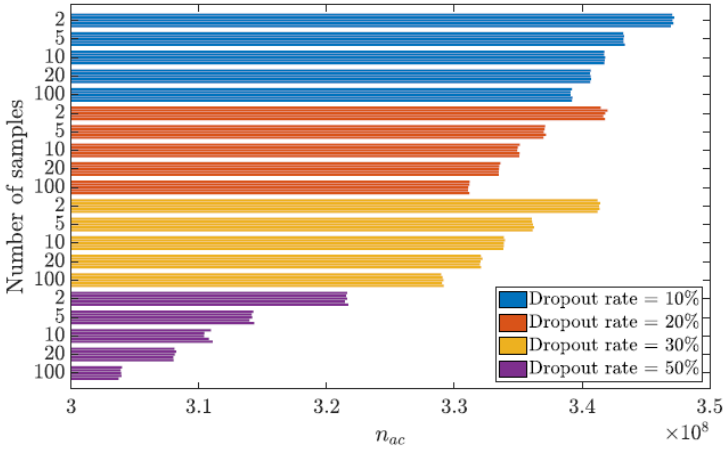
- Due to randomness: repeat all experiments 15 times
  - Final results: Mean and standard deviation of the each evaluation metric

- Comparison of prediction for different dropout rates:
  - Model with dropout rate 0%
  - Model with default setting

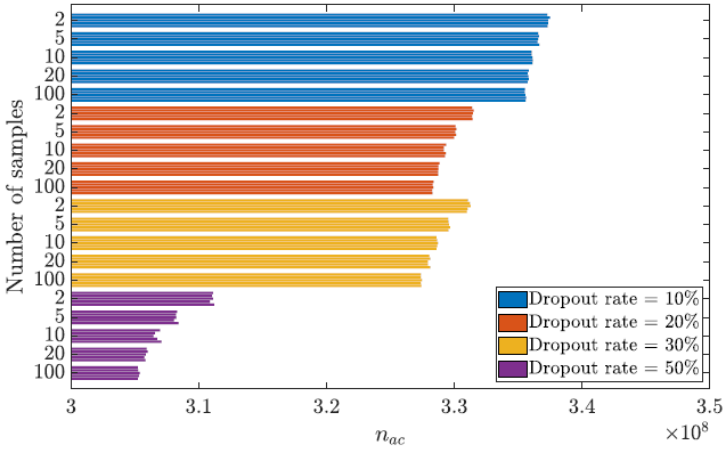
Part 2: Influence of Sample Size

Sample size	Metrics based on standard deviation p(accurate|certain)	p(uncertain|inaccurate)	PAvPU	Metrics based on entropy p(accurate|certain)	p(uncertain|inaccurate)	PAvPU	Runtime (ms)
	dropout rate = 10%			dropout rate = 10%			
2 samples	98.88% ± 0.01%	74.09% ± 0.15%	92.41% ± 0.03%	99.49% ± 0.01%	88.14% ± 0.08%	90.48% ± 0.03%	88
5 samples	99.28% ± 0.01%	81.76% ± 0.11%	91.74% ± 0.02%	99.53% ± 0.01%	88.60% ± 0.07%	90.25% ± 0.02%	198
10 samples	99.34% ± 0.01%	83.00% ± 0.07%	91.42% ± 0.02%	99.55% ± 0.01%	88.83% ± 0.05%	90.14% ± 0.02%	388
20 samples	99.38% ± 0.01%	83.64% ± 0.07%	91.17% ± 0.02%	99.56% ± 0.01%	88.96% ± 0.04%	90.07% ± 0.02%	755
100 samples	99.43% ± 0.00%	84.44% ± 0.03%	90.82% ± 0.01%	99.57% ± 0.00%	89.12% ± 0.03%	90.01% ± 0.01%	3739
	dropout rate = 20%			dropout rate = 20%			
2 samples	98.25% ± 0.02%	73.16% ± 0.20%	91.57% ± 0.05%	99.09% ± 0.01%	87.94% ± 0.11%	89.68% ± 0.04%	83
5 samples	98.78% ± 0.01%	81.70% ± 0.15%	90.78% ± 0.05%	99.19% ± 0.01%	88.77% ± 0.09%	89.35% ± 0.04%	204
10 samples	98.93% ± 0.01%	83.27% ± 0.11%	90.37% ± 0.03%	99.24% ± 0.01%	89.11% ± 0.08%	89.17% ± 0.03%	387
20 samples	99.01% ± 0.01%	84.14% ± 0.08%	90.02% ± 0.02%	99.27% ± 0.00%	89.32% ± 0.05%	89.03% ± 0.02%	750
100 samples	99.12% ± 0.01%	85.23% ± 0.03%	89.50% ± 0.02%	99.30% ± 0.00%	89.55% ± 0.02%	88.93% ± 0.01%	3666
	dropout rate = 30%			dropout rate = 30%			
2 samples	98.49% ± 0.02%	74.39% ± 0.28%	91.54% ± 0.04%	99.28% ± 0.01%	89.02% ± 0.13%	89.58% ± 0.04%	82
5 samples	99.04% ± 0.01%	83.35% ± 0.17%	90.56% ± 0.04%	99.39% ± 0.01%	89.95% ± 0.09%	89.15% ± 0.04%	200
10 samples	99.20% ± 0.01%	85.09% ± 0.09%	90.03% ± 0.03%	99.45% ± 0.01%	90.42% ± 0.08%	88.90% ± 0.02%	380
20 samples	99.28% ± 0.01%	86.09% ± 0.10%	89.62% ± 0.03%	99.48% ± 0.01%	90.70% ± 0.07%	88.75% ± 0.03%	750
100 samples	99.39% ± 0.01%	87.36% ± 0.07%	88.96% ± 0.02%	99.51% ± 0.00%	90.99% ± 0.04%	88.61% ± 0.02%	3690
	dropout rate = 50%			dropout rate = 50%			
2 samples	95.84% ± 0.03%	66.93% ± 0.33%	88.12% ± 0.04%	97.48% ± 0.03%	83.34% ± 0.13%	86.93% ± 0.03%	104
5 samples	96.86% ± 0.05%	77.47% ± 0.26%	87.17% ± 0.08%	97.70% ± 0.03%	85.02% ± 0.18%	86.40% ± 0.07%	214
10 samples	97.17% ± 0.03%	80.04% ± 0.18%	86.53% ± 0.07%	97.81% ± 0.02%	85.80% ± 0.13%	86.08% ± 0.06%	453
20 samples	97.37% ± 0.02%	81.65% ± 0.09%	85.99% ± 0.04%	97.89% ± 0.01%	86.32% ± 0.07%	85.87% ± 0.04%	749
100 samples	97.67% ± 0.02%	83.99% ± 0.11%	85.14% ± 0.04%	97.96% ± 0.01%	86.85% ± 0.06%	85.77% ± 0.03%	3747

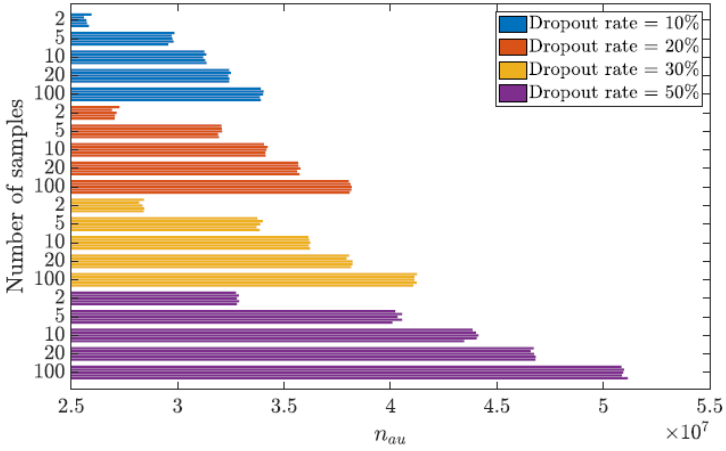
Part 2: Influence of Sample Size



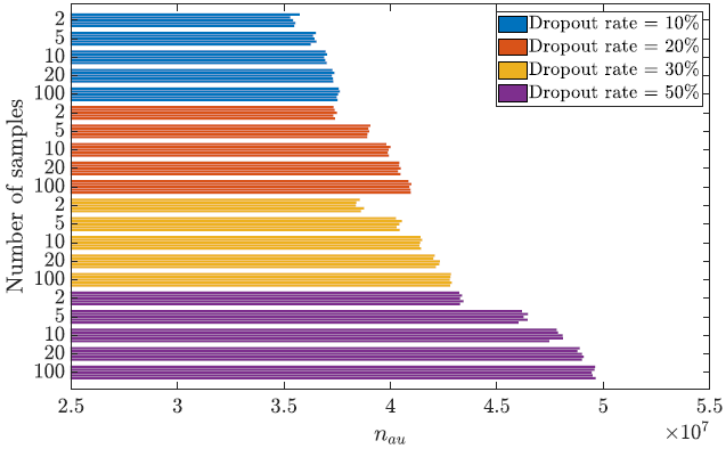
(a) n_{ac} for standard deviation.



(b) n_{ac} for entropy.



(c) n_{au} for standard deviation.



(d) n_{au} for entropy.

Part 2: Influence of Dropout Rate

- Impact of Dropout Rates on Mean IoU and ECE:
  - **Mean IoU:** Lower dropout rates → higher mean IoU values, with 0% dropout rate showing the highest performance
  - **ECE:** Best ECE with a dropout rate of 0%, difference of ECE minimal for dropout rates ranging from 10% to 30%

dropout rate	$p_{ac,\text{std}}$	$p_{ui,\text{std}}$	$\text{PAvPU}_{\text{std}}$	$p_{ac,\text{en}}$	$p_{ui,\text{en}}$	PAvPU_{en}	mean IoU	ECE
0%	-	-	-	-	-	-	80.09%	1.98%
default	-	-	-	-	-	-	79.40%	4.29%
10%	99.38%	83.64%	91.17%	99.56%	88.96%	90.07%	77.25%	5.62%
20%	99.01%	84.14%	90.02%	99.27%	89.32%	89.03%	76.16%	6.06%
30%	99.28%	86.09%	89.62%	99.48%	90.70%	88.75%	75.34%	5.42%
50%	97.37%	81.65%	85.99%	97.89%	86.32%	85.87%	70.06%	8.15%

Part 2: Influence of Dropout Rate

- Impact of Dropout Rates on Uncertainty Quality:
  - $p(\mathbf{accurate}|\mathbf{certain})$: approach 100%, differences minimal between dropout rates.
  - $p(\mathbf{uncertain}|\mathbf{inaccurate})$: range from 80% to 90%, changes are more noticeable.
  - PAvPU: decrease gradually with higher dropout rates.

dropout rate	$p_{ac,\text{std}}$	$p_{ui,\text{std}}$	$\text{PAvPU}_{\text{std}}$	$p_{ac,\text{en}}$	$p_{ui,\text{en}}$	PAvPU_{en}	mean IoU	ECE
0%	-	-	-	-	-	-	80.09%	1.98%
default	-	-	-	-	-	-	79.40%	4.29%
10%	99.38%	83.64%	91.17%	99.56%	88.96%	90.07%	77.25%	5.62%
20%	99.01%	84.14%	90.02%	99.27%	89.32%	89.03%	76.16%	6.06%
30%	99.28%	86.09%	89.62%	99.48%	90.70%	88.75%	75.34%	5.42%
50%	97.37%	81.65%	85.99%	97.89%	86.32%	85.87%	70.06%	8.15%

Part 2: Influence of Dropout Rate

- Based on $p(\mathbf{accurate}|\mathbf{certain})$ and PAvPU: 10% dropout rate performs the best
- Based on $p(\mathbf{uncertain}|\mathbf{inaccurate})$: 30% dropout rate performs the best
- Differences in performance between 10% and 30% dropout rates are more noticeable for $p(\mathbf{uncertain}|\mathbf{inaccurate})$.
- $p(\mathbf{uncertain}|\mathbf{inaccurate})$: more important in the industrial context
  - helps the identification of potentially incorrect predictions through uncertainty estimation
- Overall: 30% dropout rate is optimal for uncertainty estimation, 50% dropout rate performs worst

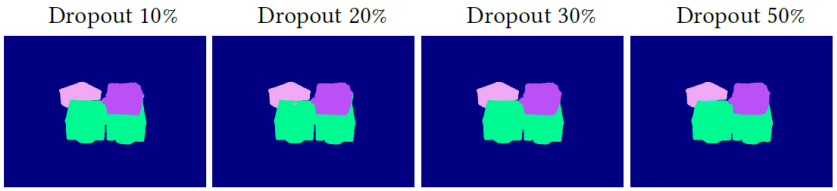
dropout rate	$p_{ac,\text{std}}$	$p_{ui,\text{std}}$	$\text{PAvPU}_{\text{std}}$	$p_{ac,\text{en}}$	$p_{ui,\text{en}}$	PAvPU_{en}	mean IoU	ECE
0%	-	-	-	-	-	-	80.09%	1.98%
default	-	-	-	-	-	-	79.40%	4.29%
10%	99.38%	83.64%	91.17%	99.56%	88.96%	90.07%	77.25%	5.62%
20%	99.01%	84.14%	90.02%	99.27%	89.32%	89.03%	76.16%	6.06%
30%	99.28%	86.09%	89.62%	99.48%	90.70%	88.75%	75.34%	5.42%
50%	97.37%	81.65%	85.99%	97.89%	86.32%	85.87%	70.06%	8.15%

Part 2: Influence of Uncertainty Metrics

- Based on $p(\mathbf{accurate}|\mathbf{certain})$ and $p(\mathbf{uncertain}|\mathbf{inaccurate})$: entropy are better
- Based on PAvPU: standard deviation are better
- the differences in PAvPU values are minimal (<1.1%)
- The differences in $p(\mathbf{uncertain}|\mathbf{inaccurate})$ are noticeable (~5%)
- → Entropy is more suitable

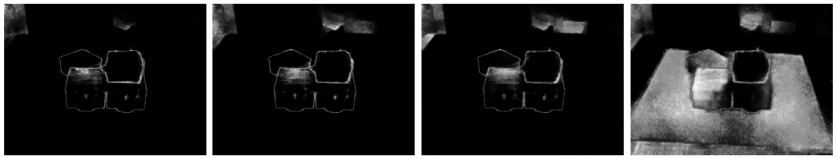
dropout rate	$p_{ac,\text{std}}$	$p_{ui,\text{std}}$	$\text{PAvPU}_{\text{std}}$	$p_{ac,\text{en}}$	$p_{ui,\text{en}}$	PAvPU_{en}
0%	-	-	-	-	-	-
default	-	-	-	-	-	-
10%	99.38%	83.64%	91.17%	99.56%	88.96%	90.07%
20%	99.01%	84.14%	90.02%	99.27%	89.32%	89.03%
30%	99.28%	86.09%	89.62%	99.48%	90.70%	88.75%
50%	97.37%	81.65%	85.99%	97.89%	86.32%	85.87%

Part 2: Uncertainty Estimation

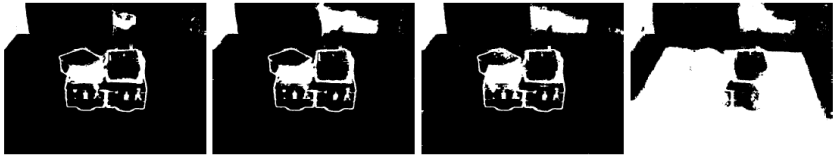


(a) Predictions.

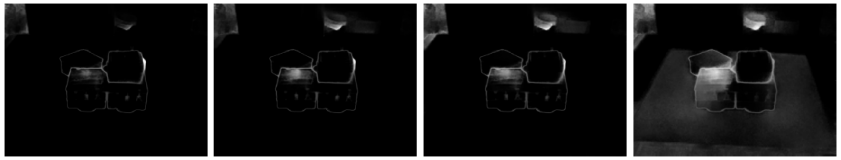
(b) Binary accuracy maps.



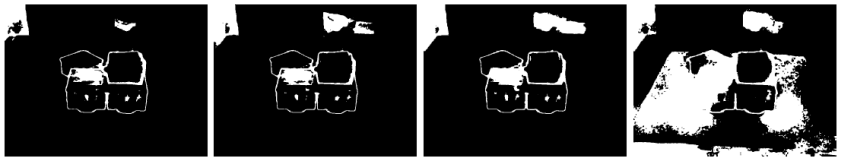
(c) Standard deviation maps.



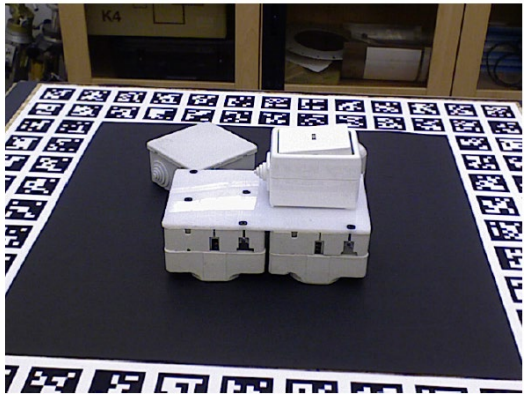
(d) Binary uncertainty maps based on standard deviation.



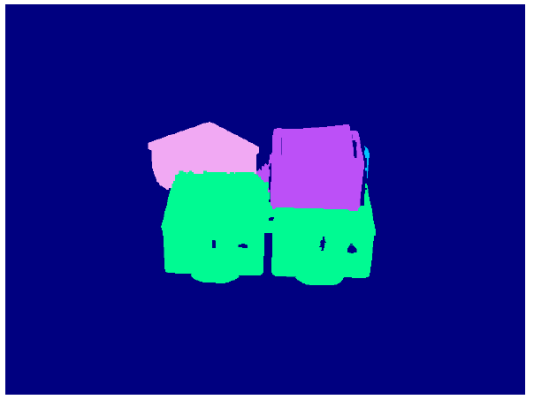
(f) Entropy maps.



(g) Binary uncertainty maps based on entropy.



(a) Scene 4, image 325.



(b) Ground truth.

Conclusion

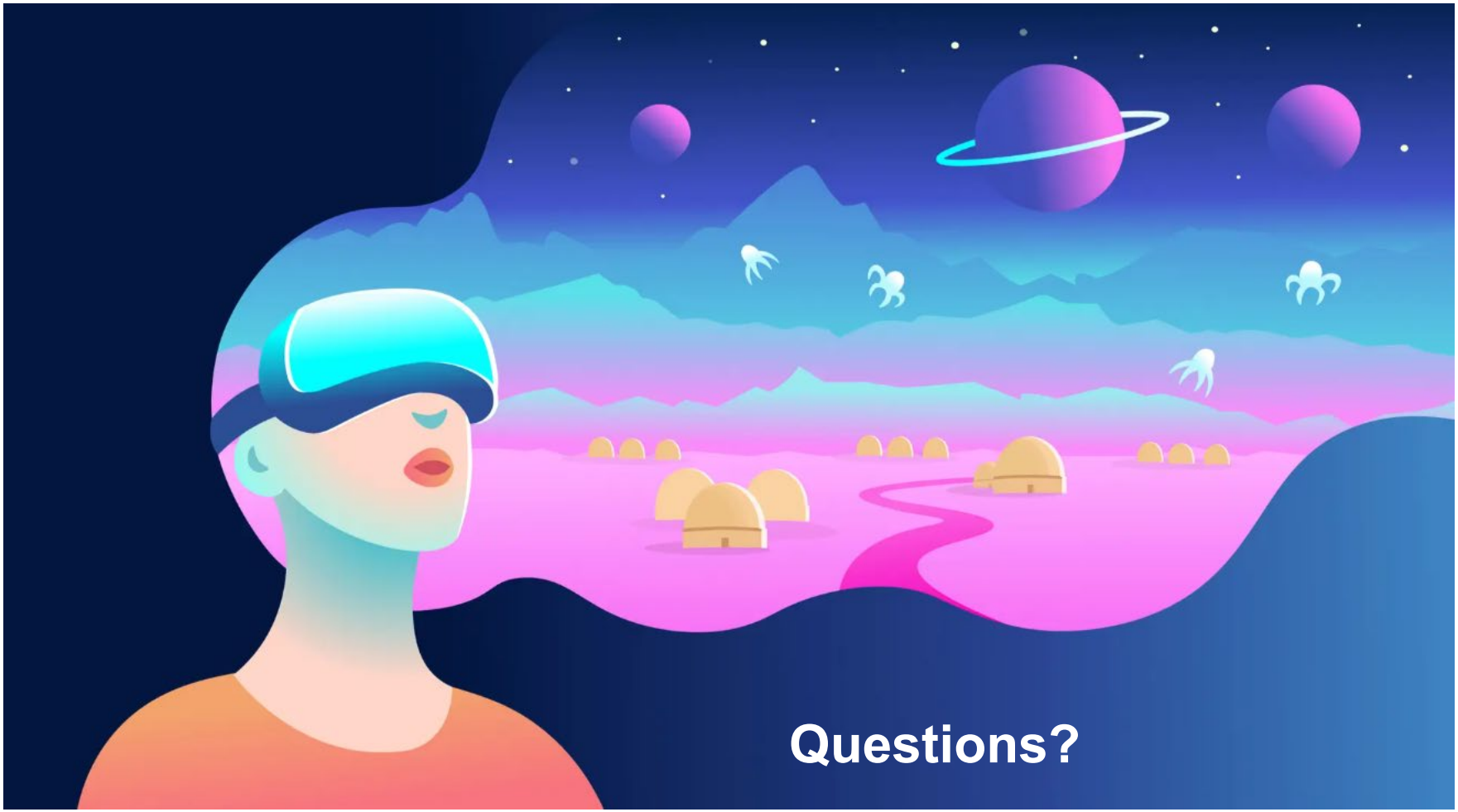
- the performance of semantic segmentation and the uncertainty estimation of the ViT (SegFormer) in an industrial context are evaluated

- Different training configurations and hyperparameters tested, focusing on the semantic segmentation performance:
  - Training datasets
  - Model backbones
  - Data augmentation
  - Learning Rates

- Segformer is combined with MC Dropout with different configurations, focusing on the quality of the uncertainty estimation
  - Sample sizes
  - Dropout rate
  - Uncertainty metric

Outlook

- Integrate uncertainty estimation into training process / post-processing step
- Expand testing to other datasets of industrial context
  - LineMOD
  - YCB-Video
- Incorporate instance segmentation for pose estimation
  - Mask2Former
  - DETR

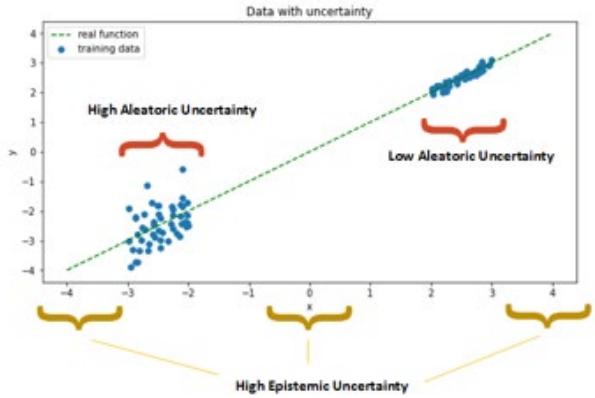

Questions?

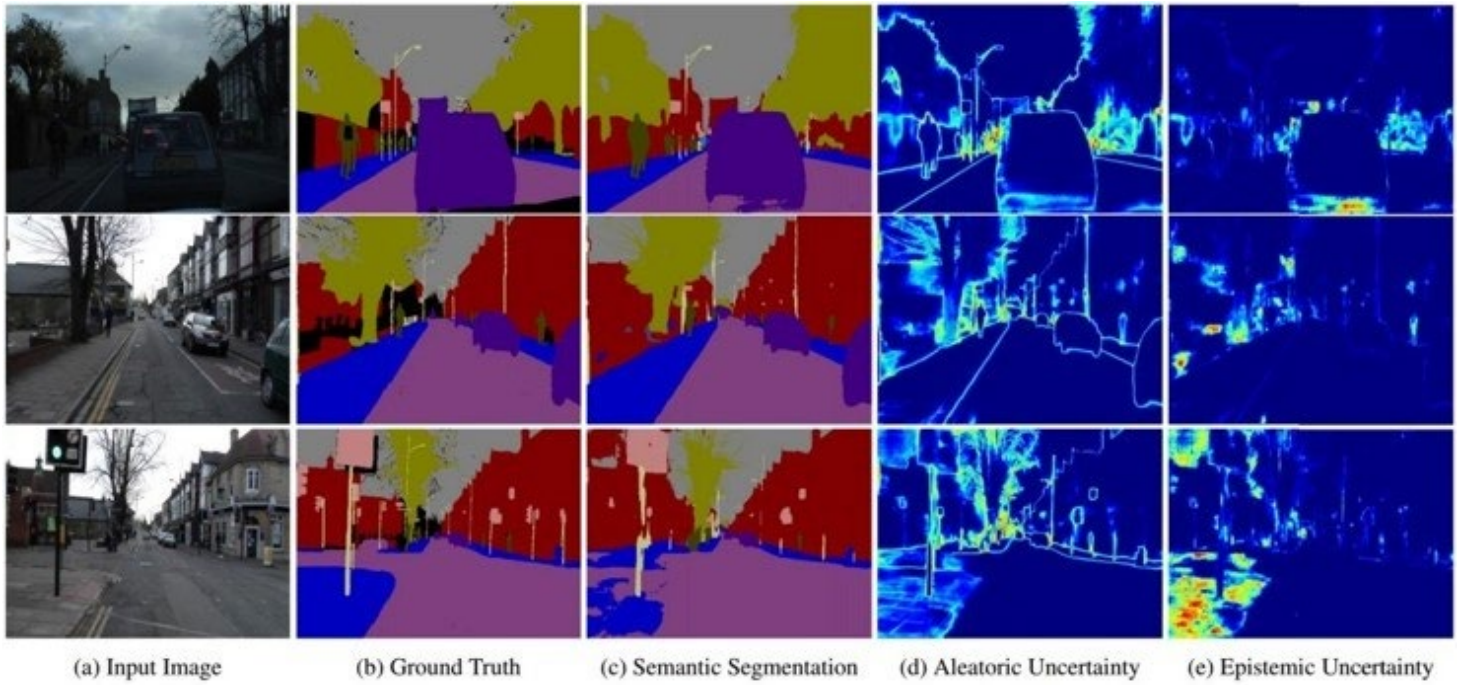
Literatur

- [1] Naresh, Y. G., Suzanne Little and Noel E. O'Connor. "A Residual Encoder-Decoder Network for Semantic Segmentation in Autonomous Driving Scenarios." 2018 26th European Signal Processing Conference (EUSIPCO) (2018): 1052-1056.
- [2] Sundelius, Carl. "Deep Fusion of Imaging Modalities for Semantic Segmentation of Satellite Imagery." (2018).
- [3] Holste, Gregory, Ryan, Sullivan, Michael,Bindschadler, Nicholas, Nagy, and Adam, Alessio. "Multi-class semantic segmentation of pediatric chest radiographs." . In Medical Imaging 2020: Image Processing (pp. 113131E) (2020).
- [4] https://paperswithcode.com/sota/semantic-segmentation-on-ade20k (Letzter Zugriff: 04.04.2024)
- [5] Mukhoti, Jishnu, and Yarin Gal. "Evaluating bayesian deep learning methods for semantic segmentation." arXiv preprint arXiv:1811.12709 (2018).
- [5] A. Kirillov, K. He, R. Girshick, C. Rother and P. Dollár, "Panoptic Segmentation," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 9396-9405, doi: 10.1109/CVPR.2019.00963.
- [5] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
- [5] Van Katwyk, P., Fox-Kemper, B., Seroussi, H., Nowicki, S., & Bergen, K. J. (2023). A variational LSTM emulator of sea level contribution from the Antarctic ice sheet. Journal of Advances in Modeling Earth Systems, 15, e2023MS003899. https://doi.org/10.1029/2023MS003899
- [5] Yarin Gal and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In: international conference on machine learning. PMLR. 2016, pp. 1050–1059.
- [5] Jishnu Mukhoti and Yarin Gal. "Evaluating Bayesian Deep Learning Methods for Semantic Segmentation". In: arXiv e-prints (Nov. 2018). arXiv: 1811.12709 [cs.CV].
- [6] T. Hodaň, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, X. Zabulis, T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-less Objects, IEEE Winter Conference on Applications of Computer Vision (WACV), 2017, Santa Rosa,
- [7] Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020).
- [8] Xie, Enze, et al. "SegFormer: Simple and efficient design for semantic segmentation with transformers." Advances in Neural Information Processing Systems 34 (2021): 12077-12090.

Uncertainty

- **epistemic uncertainty** (knowledge uncertainty):
  - caused by limited data and knowledge
  - can be reduced through appropriate training data/model
- **aleatoric uncertainty** (data uncertainty):
  - arises from the natural stochasticity of observations
  - cannot be mitigated even when more data is provided





Uncertainty Estimation: Monte Carlo Dropout

- variational inference
- Given:
  - training samples: $\mathbf{X}$
  - corresponding labels: $\mathbf{Y}$
  - a full deep Gaussian Process over the weights: $\mathbf{W}$
- Goal: obtain a posterior distribution $p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$
- Difficulty: many parameters & non-linear activation functions
- → Approximation of the posterior distribution: the variational distribution $q(\mathbf{W})$
- → Kullback-Leibler (KL) divergence: $\mathrm{KL}(q(\mathbf{W})|p(\mathbf{W}|\mathbf{X}, \mathbf{Y}))$
- $q(\mathbf{W})$ follows a Bernoulli distribution
- Bernoulli distribution with parameter p_b = dropout layer with a dropout rate of p_b
- minimize the cross-entropy loss function using standard optimization algorithms → reduction in the KL divergence term

Self-attention

- Establishes relationships between different elements in an input sequence
- Allows model to consider all inputs simultaneously

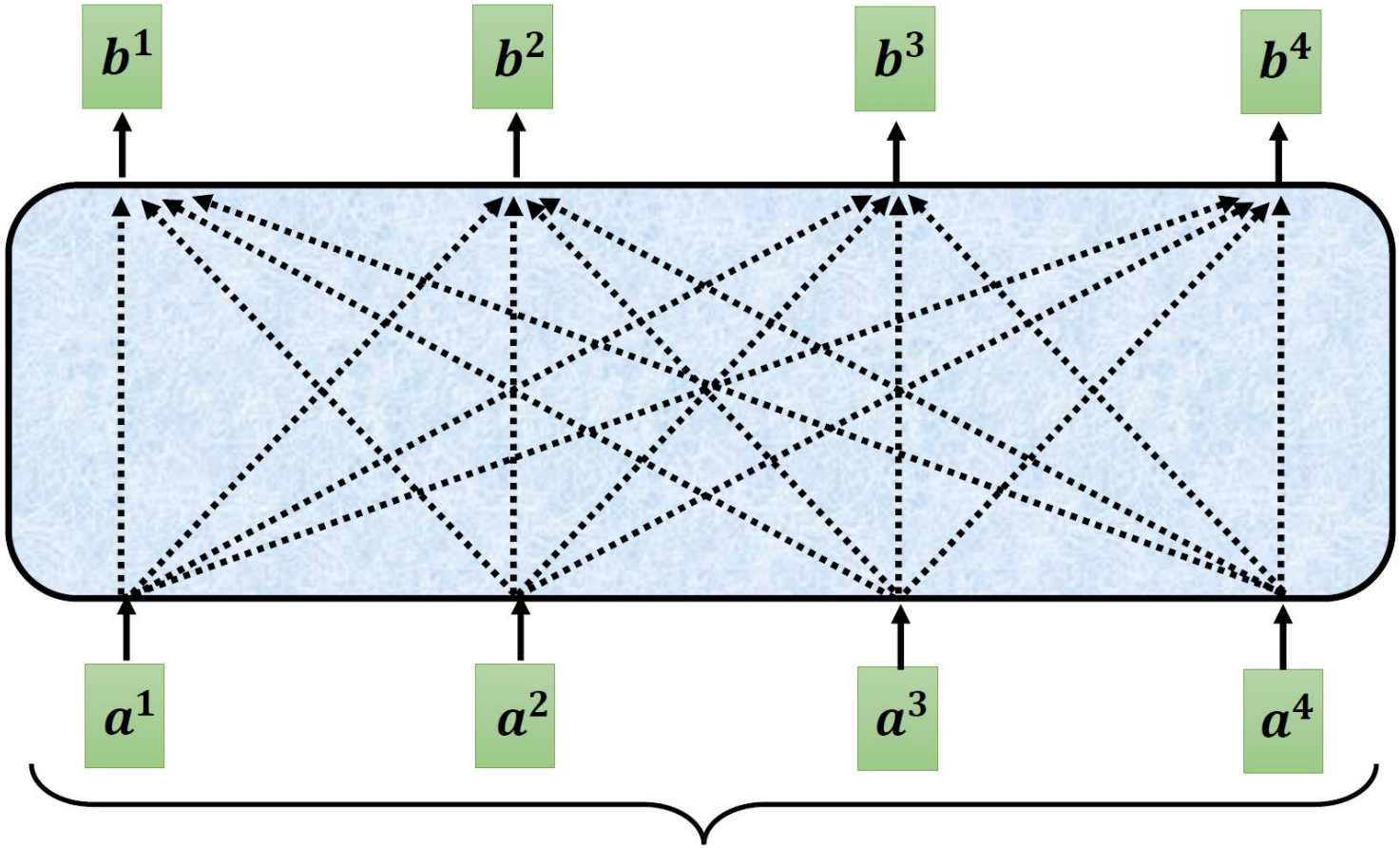
- Query: $Q = W^q I$
- Key: $K = W^k I$
- Value: $V = W^v I$

→ Parameters to be learned

- Attention matrix: $A' \xleftarrow{\text{softmax}} = A = K^T \cdot Q$
- Weighted sum: $O = V \cdot A'$

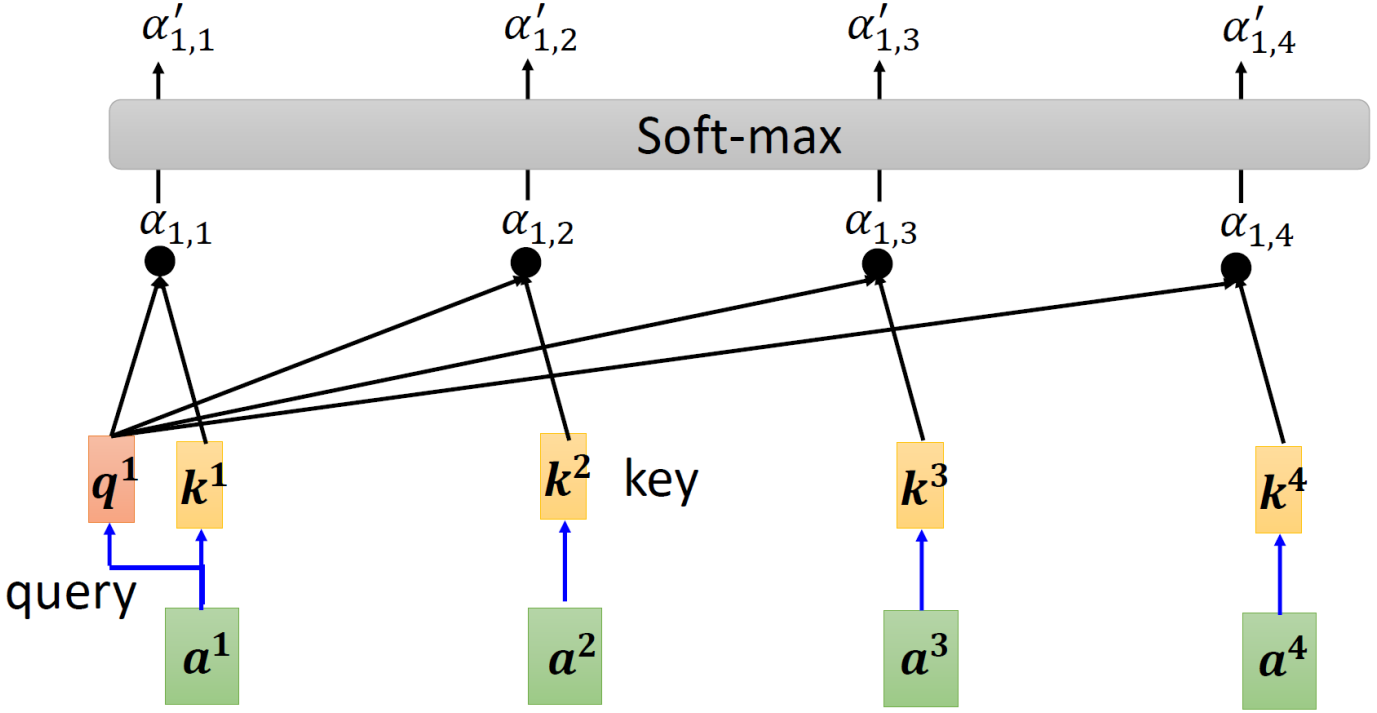
- Multi-head Self-attention: different types of relevance & correlations
- Advantage:
  - able to capture long-range dependencies
  - Comprehend global informations

Self-attention



Self-attention

$$\alpha'_{1,i} = exp(\alpha_{1,i}) / \sum_j exp(\alpha_{1,j})$$



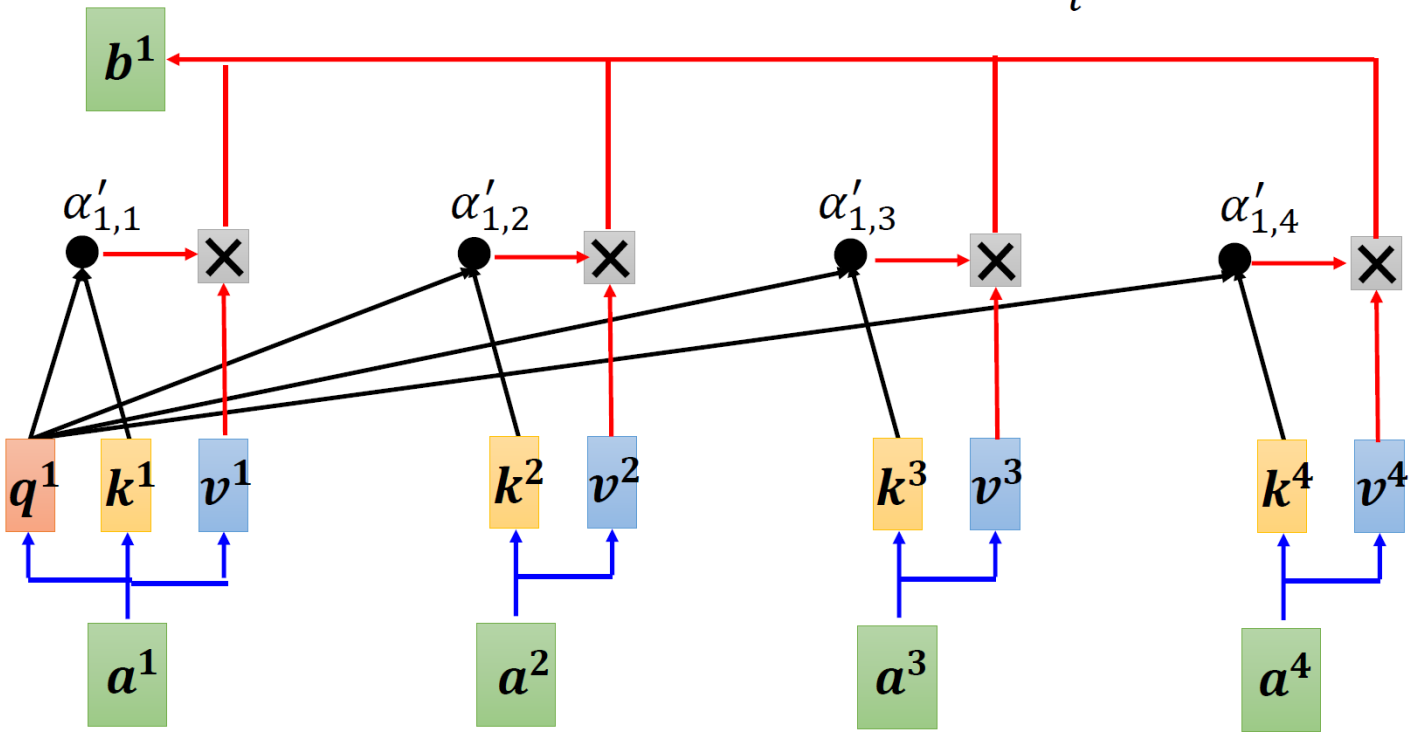
$$q^1 = W^q a^1 \qquad k^2 = W^k a^2 \qquad k^3 = W^k a^3 \qquad k^4 = W^k a^4$$

$$k^1 = W^k a^1$$

Self-attention

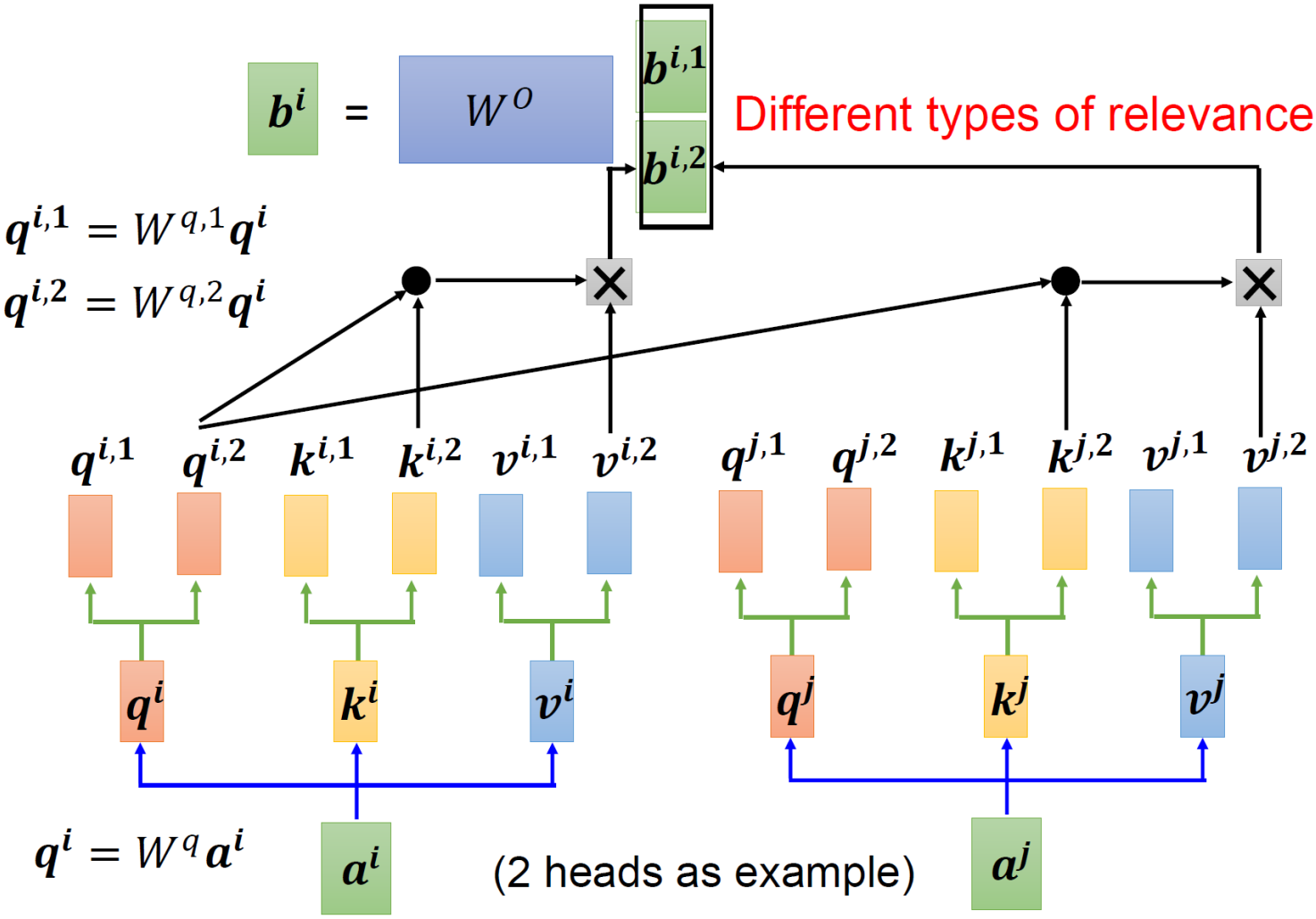
Extract information based on attention scores

$$b^1 = \sum_i \alpha'_{1,i} v^i$$



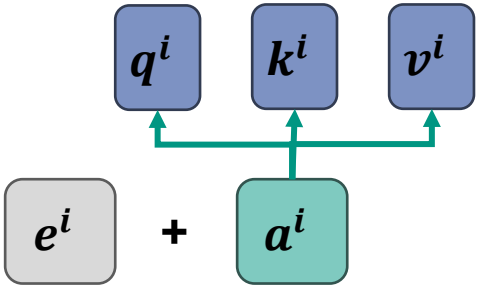
$$v^1 = W^v a^1 \qquad v^2 = W^v a^2 \qquad v^3 = W^v a^3 \qquad v^4 = W^v a^4$$

Multi-head Self-attention

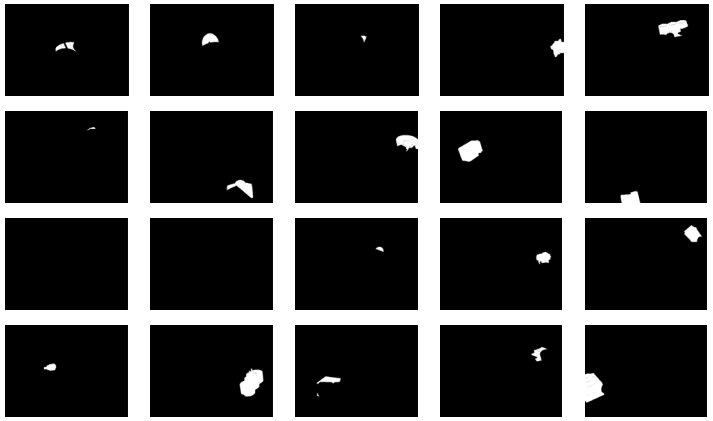


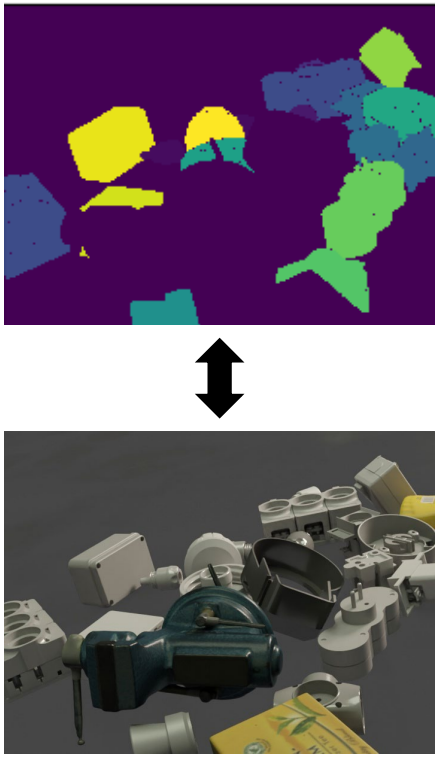
Self-attention: Positional Encoding

- No position information in self-attention
- Each position has a unique positional vector e^i
  - **Handcrafted**
  - **Learned from data**



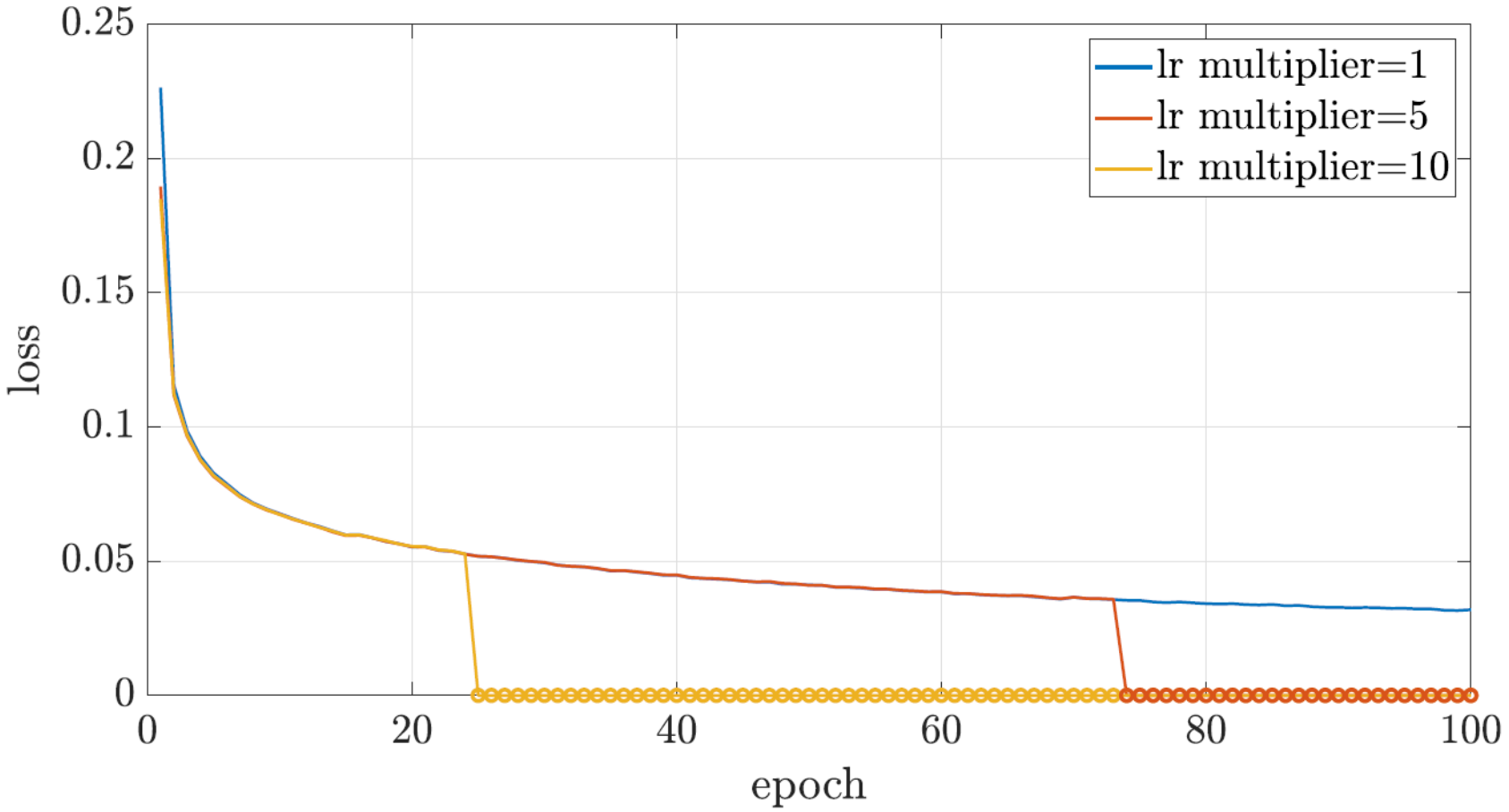
Data preprocessing





- Generate one label image from multiple segmentation masks
  - Assigning ID to each pixel
  - Combining all pixels

Implementation Details: Learning Rate Multiplier

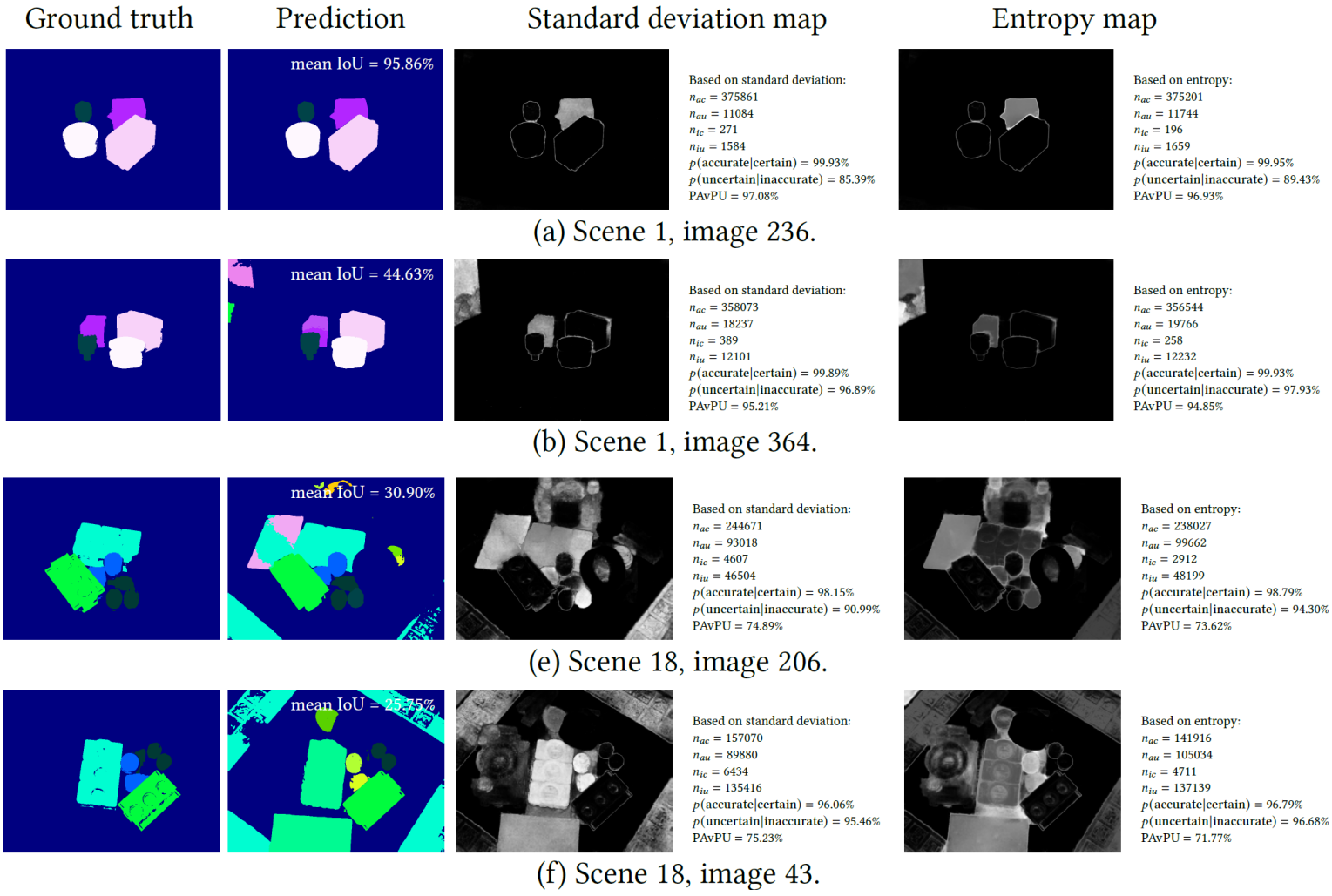


Uncertainty Estimation: Approximation

- Ideal: a dropout layer after every hidden layer
- Disadvantage:
  - → a large neural network → slows down training
  - a large amount of work to modifying the model

- Therefore, use the dropout layers that are already integrated
  - in the SegFormer encoder after each efficient self-attention layer
  - after each linear layer in the attention output layer
  - after each Mix-FFN Layer.
  - one in the SegFormer decoder.

Part 2: Uncertainty Estimation



SegFormer: model size

Table 3.1: Mix Transformer encoders in different sizes [31].

Model variant	Depths	Hidden sizes	Decoder hidden size	Params (Million)
MiT-B0	[2, 2, 2, 2]	[32, 64, 160, 256]	256	3.7
MiT-B1	[2, 2, 2, 2]	[64, 128, 320, 512]	256	14.0
MiT-B2	[3, 4, 6, 3]	[64, 128, 320, 512]	768	25.4
MiT-B3	[3, 4, 18, 3]	[64, 128, 320, 512]	768	45.2
MiT-B4	[3, 8, 27, 3]	[64, 128, 320, 512]	768	62.6
MiT-B5	[3, 6, 40, 3]	[64, 128, 320, 512]	768	82.0