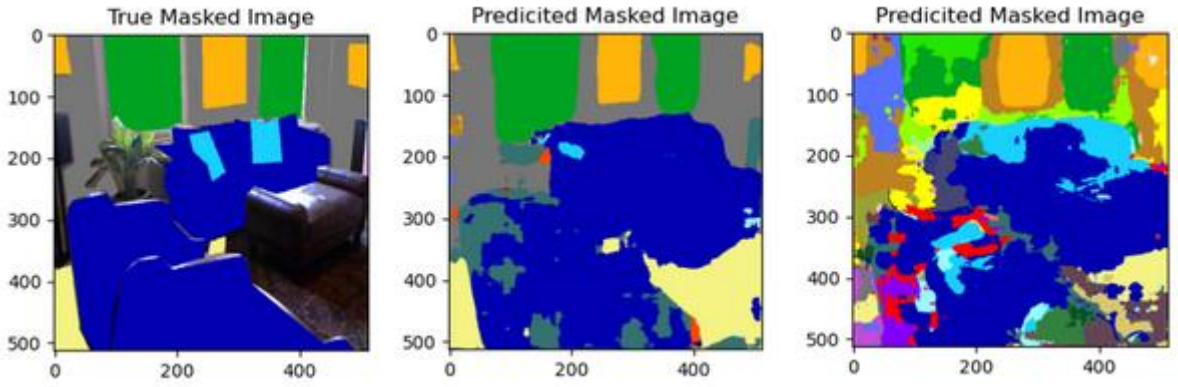
Semantic segmentation with convolutional deep encoder-decoder networks on SUN RGB data

25.01.2024

Mikolaj Antoni Baranski
Alisia Marianne Michel



Agenda

1. Introduction
2. Data overview
3. Model design
4. Methodology
5. Results
6. Further research

1. Introduction

Semantic segmentation has been approached with various methods like convolution and tranformer-based approaches as well as multi-modal models.

Research context

Models for semantic segmentation

Project Focus

Fully convolutional networks

- *Based on CNNs*
- *Local context windows*
- *Examples: U-Net (2015) and ConvNeXt (2022)*

Transformer-based models

- *Applies self-attention possibly limited by shifted windows*
- *Global context information*
- *Example: Swin Transformer (2021)*

Multi-modal models

- *Combination of different types of data sources*
- *New task of zero-shot semantic segmentation*
- *Example: ZS3Net (2019)*

Performance Measure

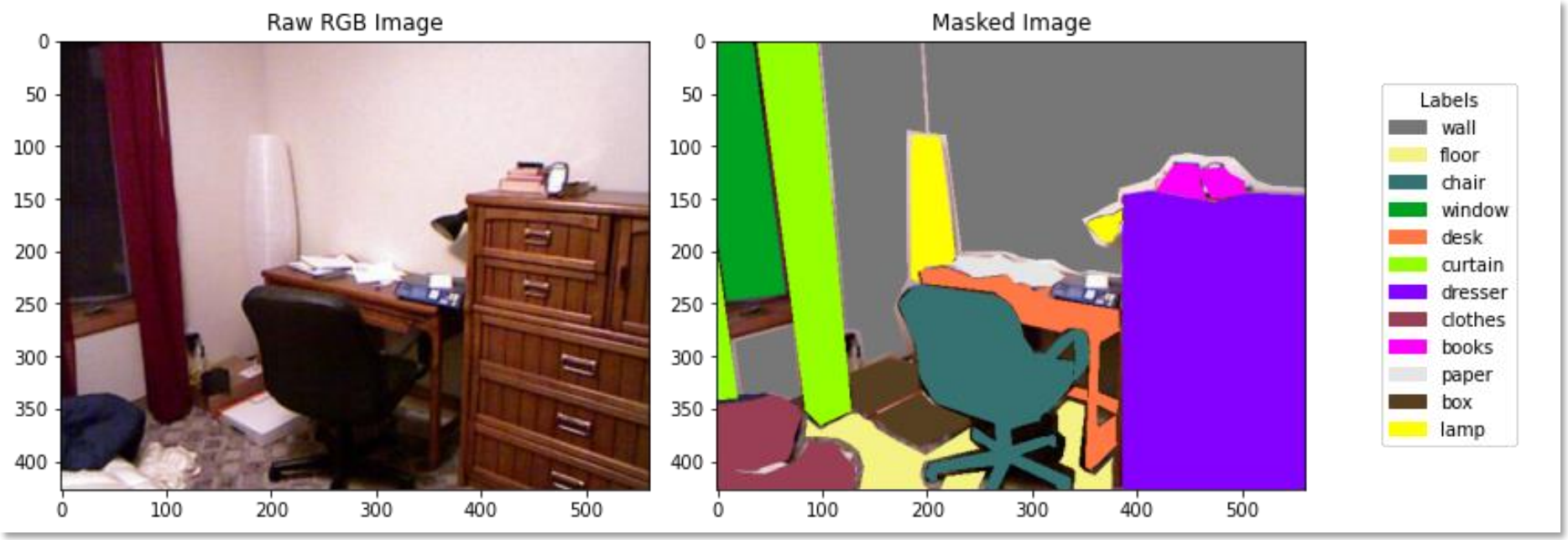
$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Average IoU/ Jaccard index over all object classes

2. Data overview

We train and test our models on the SUN-RGBD dataset composed of 10k images of indoor scenes with rich label masks – we focus on a subset of 37 classes

Data overview



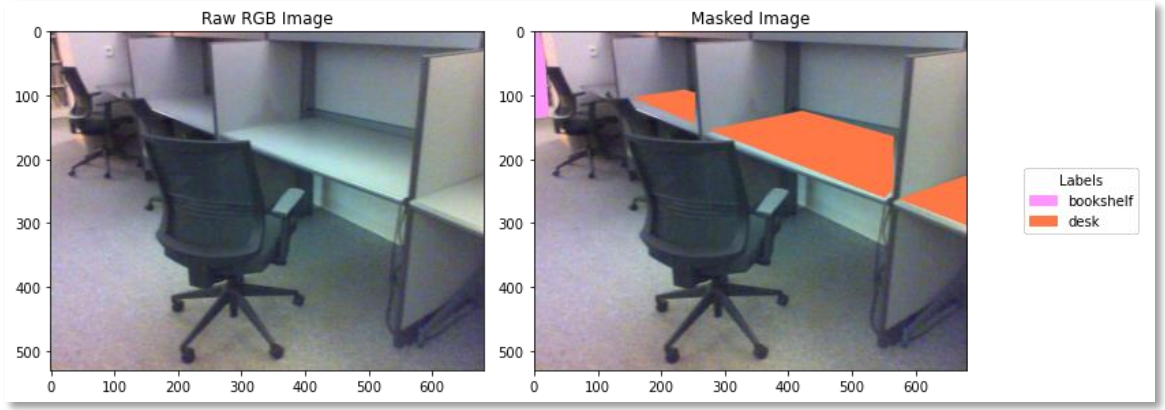
- **37 classes** masking **10,335 images** of various indoor scenes – we only use the **RGB channels**
- Photographs from past datasets relabeled and verified by the authors

We train and test our models on the SUN-RGBD dataset composed of 10k images of indoor scenes with rich label masks – we focus on a subset of 37 classes

Data overview

1 **Varying label quality**

- *The images have **widely varying quality of labels***
- *Errors include mislabels, empty image portions, or poor polygons*



2 **Imbalance of labels**

- *The dataset, even when reduced to 37 classes has a high class imbalance*
- *While, wall and floor are good to be highly represented due to their occurrence in all images, other classes are difficult for models to learn*

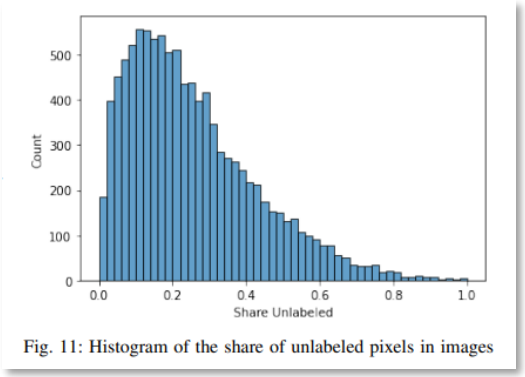
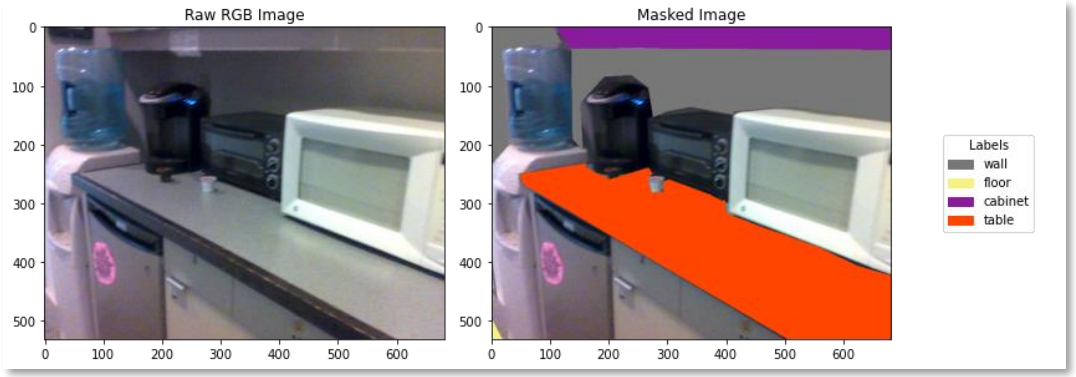


Fig. 11: Histogram of the share of unlabeled pixels in images



We train and test our models on the SUN-RGBD dataset composed of 10k images of indoor scenes with rich label masks – we focus on a subset of 37 classes

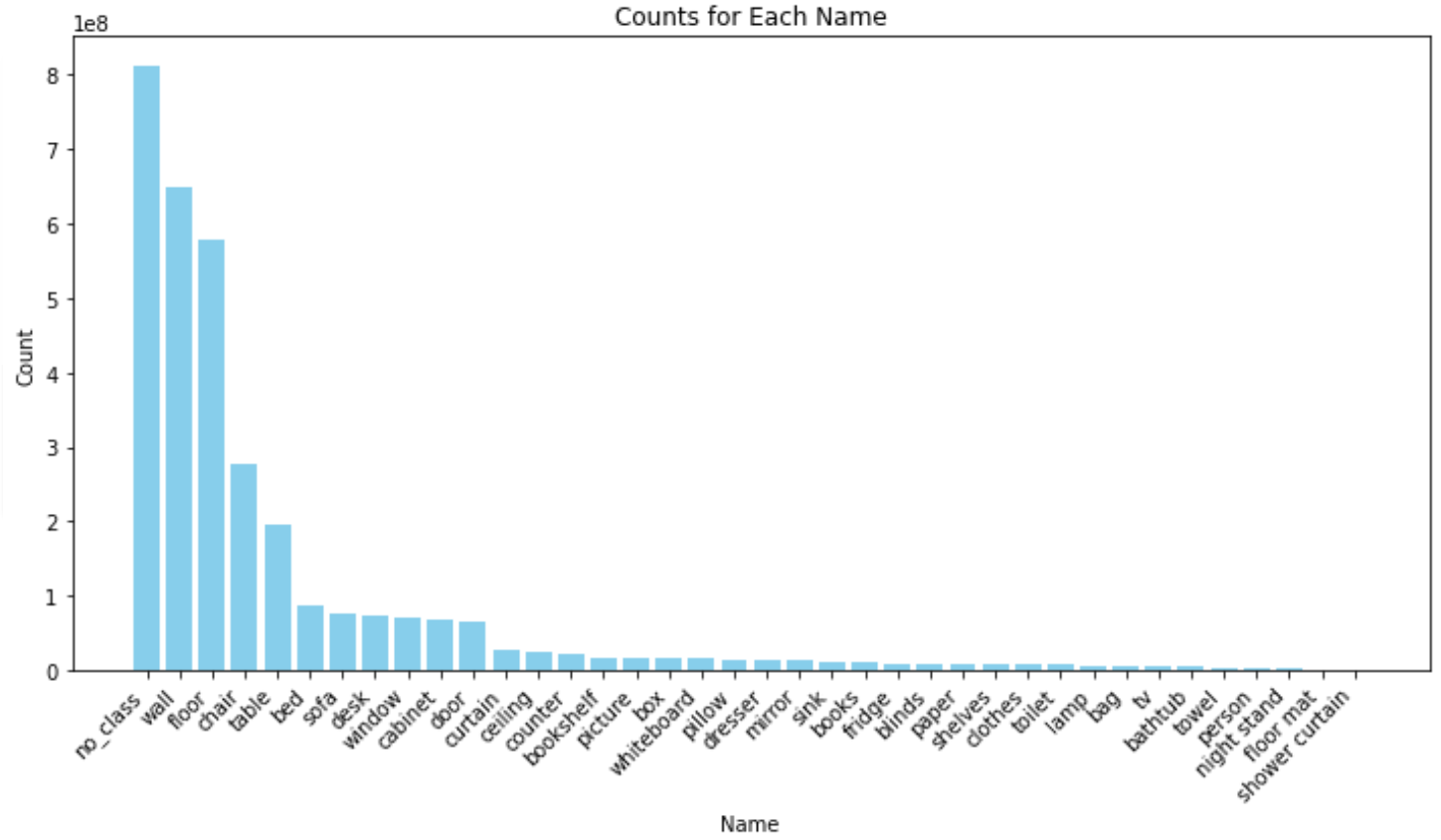
Data overview

1 **Varying label quality**

- *The images have widely varying quality of labels*
- *Errors include mislabels, empty image portions, or poor polygons*

2 **Imbalance of labels**

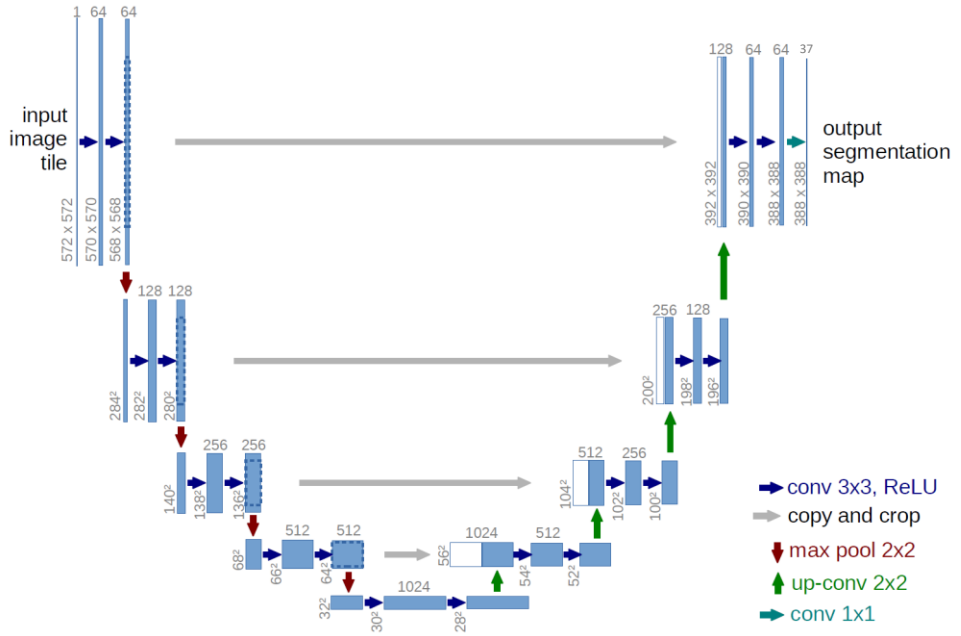
- *The dataset, even when reduced to 37 classes **contains high class imbalance***
- *While wall and floor are good to be highly represented due to their occurrence in all images, **other classes are difficult for models to learn***



We have chosen two cutting edge convolutional models, one a SOTA model from 2015 and the other a transformer based SOTA model from 2022

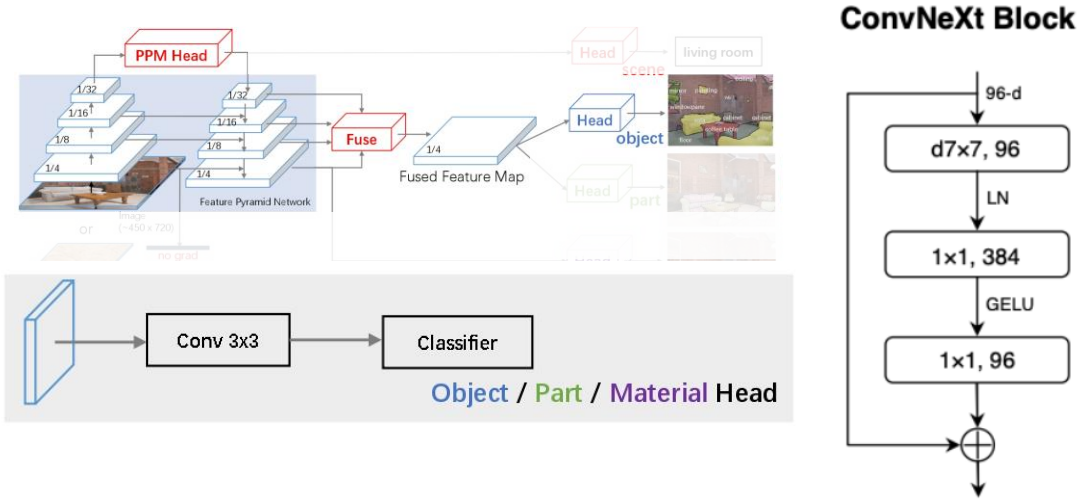
Model design

U-Net Architecture



- Model architecture based on original paper **re-implemented from scratch** (no pretraining)

ConvNeXt - UPerNet



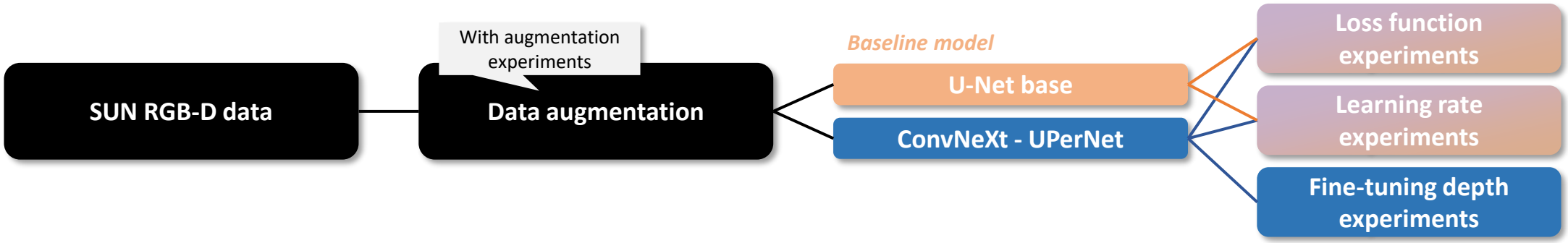
- **Pre-trained** model based on outdoor and indoor scene segmentation, fine-tuned on our dataset

3. Model design

4. Methodology

We are studying the performance of a U-Net trained from scratch and a fine-tuned SOTA model - UPerNet with ConvNeXt backbone with different settings

General approach



Experimental set--up

- Convolution based **Encoder-Decoder** model comparisons for **scene semantic segmentation**
- **37 classes** of indoor objects and scene elements
- Two models – **U-Net** (trained from scratch) and **UPerNet with a ConvNeXt** backbone (fine-tuned)
- Cross-entropy loss, max 100 epochs (with early stopping), Adam optimizer, batch size 64

Research questions:

RQ1: *"What are hyperparameter settings that improve the performance of the U-Net (and ConvNeXt) for training on the SUN RGB-D data set?"*

RQ2: *"How does the performance of a fine-tuned pretrained model compare to a model trained from scratch in semantic image segmentation?"*

RQ3: *"To what extent is the difference in performance between the models justifiable given their training costs?"*

The hyperparameters leading to the best performing U-Net on validation data are a learning rate of 0.0001 and a plateau learning rate scheduler.

U-Net training strategy

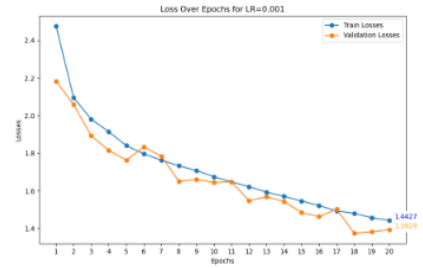
RQ1: *"What are hyperparameter settings that improve the performance of the U-Net (and ConvNeXt) for training on the SUN RGB-D data set?*

Intuition-based hyperparameter tuning

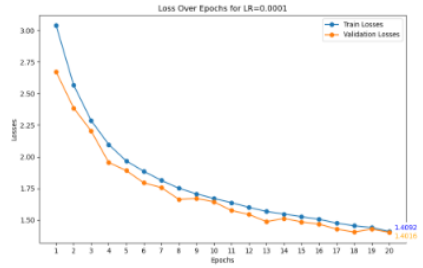
1. *Learning rate*
    1. *Learning rate values between 0.00001 and 0.01 for 20 epochs*
    2. *Learning rate scheduler: Plateau*
2. *Weighted cross-entropy loss*
3. *No data augmentation*

Cross-entropy loss, max 100 epochs (with early stopping), Adam optimizer, batch size 64

Model	Hyperparameters	Trainable parameters	Best Epoch	Val mIoU
U-Net	**Plateau + 0.0001LR**	**31,039,973**	**98**	**0.2835**
U-Net	Plateau + 0.0001LR + woAugment	31,039,973	40	0.2207
U-Net	Plateau + 0.0001LR + Weighted	31,039,973	99	0.1634



(a) Learning rate of 0.001



(b) Learning rate of 0.0001

Learning Rate	Epoch	Val Loss	Val mIoU
0.01	Nan	Nan	0.008
0.001	18	1.375	0.131
0.0001	**20**	**1.402**	**0.120**
0.00001	20	2.083	0.061
0.0001	75	1.071	0.255
0.0001 + scheduler	**73**	**0.986**	**0.279**

The hyperparameters leading to the best performing ConvNeXt on validation data are a learning rate of 0.0001 and fine-tuning depth incl. classifier and decoder

ConvNeXt fine-tuning strategy

RQ1: *"What are hyperparameter settings that improve the performance of the U-Net (and ConvNeXt) for training on the SUN RGB-D data set?*

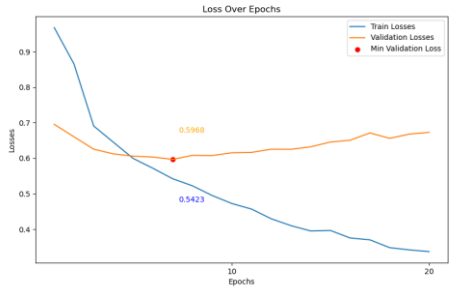
Intuition-based hyperparameter tuning

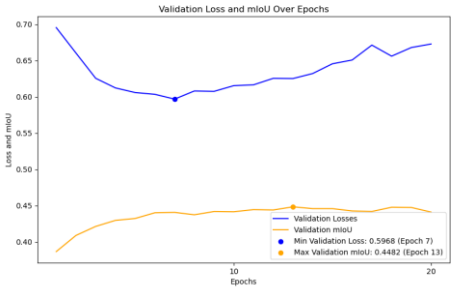
1. *Learning rate*
2. *Weighted cross-entropy loss*
3. *No data augmentation*
4. *Fine-tuning depth (Classifier, Decoder, Encoder)*

Cross-entropy loss, max 100 epochs (with early stopping), Adam optimizer, batch size 64

Model	Hyperparameters	Trainable parameters	Best Epoch	Val mIoU
ConvNeXt	Classifer + 0.001LR	28,490	13	0.4476
ConvNeXt	Classifer + 0.0001LR	28,490	92	0.4464
ConvNeXt	Decoder + 0.001LR	32,334,154	25	0.4430
ConvNeXt	**Decoder + 0.0001LR**	**32,334,154**	**13**	**0.4482**
ConvNeXt	Decoder + 0.0001LR + woAugment	32,334,154	8	0.4394
ConvNeXt	Decoder + 0.0001LR + Weighted	32,334,154	98	0.4041
ConvNeXt	Encoder + Decoder + 0.0001LR	60,155,626	18	0.4272

Training performance of top model:





5. Results

For both U-Net and CovNeXt training the model on a dataset without augmentations lead to faster convergence and lower parameter quality

Results – Augmentations

TABLE III: Training Results

Model	Hyperparameters	Trainable parameters	Best Epoch	Val mIoU
U-Net	**Plateau + 0.0001LR**	**31,039,973**	**98**	**0.2835**
U-Net	Plateau + 0.0001LR + woAugment	31,039,973	40	0.2207
U-Net	Plateau + 0.0001LR + Weighted	31,039,973	99	0.1634
ConvNeXt	Classifer + 0.001LR	28,490	13	0.4476
ConvNeXt	Classifer + 0.0001LR	28,490	92	0.4464
ConvNeXt	Decoder + 0.001LR	32,334,154	25	0.4430
ConvNeXt	**Decoder + 0.0001LR**	**32,334,154**	**13**	**0.4482**
ConvNeXt	Decoder + 0.0001LR + woAugment	32,334,154	8	0.4394
ConvNeXt	Decoder + 0.0001LR + Weighted	32,334,154	98	0.4041
ConvNeXt	Encoder + Decoder + 0.0001LR	60,155,626	18	0.4272

- Difference of **-0.06 and -0.01 for U-Net and ConvNext** respectively for the model trained on data without augmentations
- In line with expectations given **model complexity and dataset size**

For both U-Net and CovNeXt loss function weighting produced weaker results compared to the un-weighted variant – negligible difference in per category IoU

Results – loss function weighting

U-Net

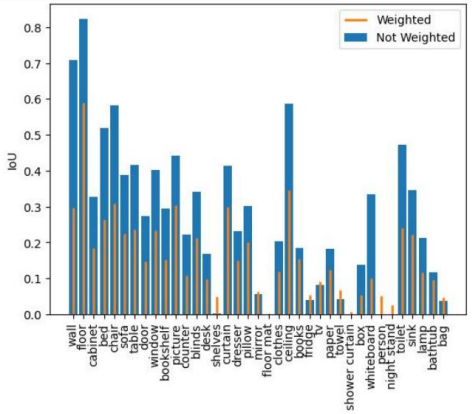


Fig. 3: U-Net IoU per class for weighted and not weighted cross-entropy loss

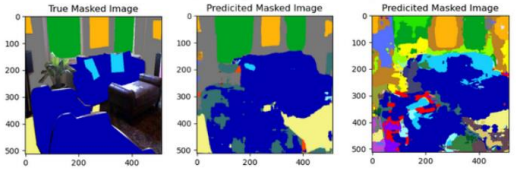


Fig. 4: Predicted mask for validation image with not weighted and weighted cross-entropy loss

ConvNeXt

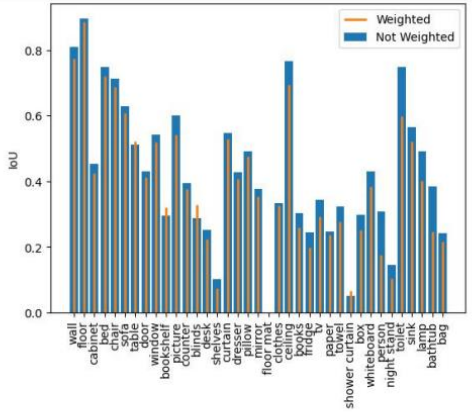


Fig. 5: ConvNeXt IoU per class for weighted and not weighted cross-entropy loss

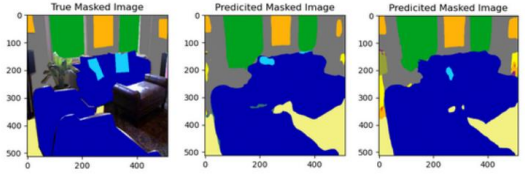


Fig. 13: Predicted mask for validation image with not weighted and weighted cross-entropy loss

Validation Results

Model	Hyperparameters	Best Epoch	Val mIoU
U-Net	**Plateau + 0.0001LR**	**98**	**0.2835**
U-Net	Plateau + 0.0001LR + woAugment	40	0.2207
U-Net	Plateau + 0.0001LR + Weighted	99	0.1634
ConvNeXt	Classifer + 0.001LR	13	0.4476
ConvNeXt	Classifer + 0.0001LR	92	0.4464
ConvNeXt	Decoder + 0.001LR	25	0.4430
ConvNeXt	**Decoder + 0.0001LR**	**13**	**0.4482**
ConvNeXt	Decoder + 0.0001LR + woAugment	8	0.4394
ConvNeXt	Decoder + 0.0001LR + Weighted	98	0.4041
ConvNeXt	Encoder + Decoder + 0.0001LR	18	0.4272

- Difference of **-0.12 and -0.04 for U-Net and ConvNext** respectively for the weighted loss function model
- Potentially a case of **underfitting**

Fine-tuning the ConvNeXt achieved highest validation mIoU when being limited to the classifer and decoder - freezing the decoder was also promising

Results – Fine-tuning Depth

Model	Hyperparameters	Trainable parameters	Best Epoch	Val mIoU
ConvNeXt	Classifer + 0.001LR	28,490	13	0.4476
ConvNeXt	**Decoder + 0.0001LR**	**32,334,154**	**13**	**0.4482**
ConvNeXt	Encoder + Decoder + 0.0001LR	60,155,626	18	0.4272

- Training the **classifier and decoder** yielded the best results
- Only retraining the classifier provided very similar results
- Fine-tuning the entire model **yielded worse** results (potentially catastrophic forgetting)

The fine-tuned ConvNeXt achieved the best results on the test data; however, both models show good level of generalizability beyond train/validation data

Results on Test data

TABLE IV: Best U-Net and ConvNeXt performance on test data

Model	Test mIoU	Val mIoU
UNet: 0.0001LR + Plateau	0.278	**0.283**
ConvNext: Decoder + 0.0001LR	**0.430**	**0.448**
Difference	*Δ0.152*	

RQ2: "How does the performance of a fine-tuned pretrained model compare to a model trained from scratch in semantic image segmentation?"

RQ3: "To what extent is the difference in performance between the models justifiable given their training costs?"

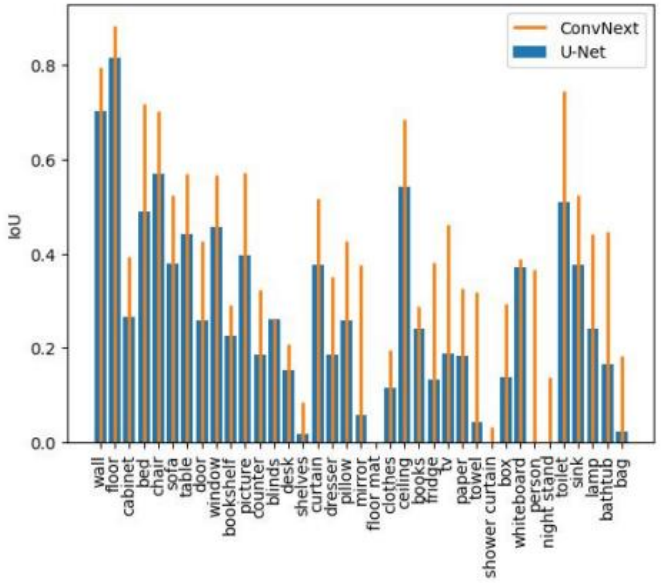


Fig. 6: IoU per class on the test set for U-Net and ConvNext

- Significantly **better performance of ConvNeXt** compared to U-Net especially on lower classes
- Test results are near validation results indicating **good generalization of both models**

The fine-tuned ConvNeXt achieved the best results on the test data; however, both models show good level of generalizability beyond train/validation data

Discussion of results

TABLE IV: Best U-Net and ConvNeXt performance on test data

Model	Test mIoU	Val mIoU
UNet: 0.0001LR + Plateau	0.278	**0.283**
ConvNext: Decoder + 0.0001LR	**0.430**	0.448
Difference	*Δ0.152*	

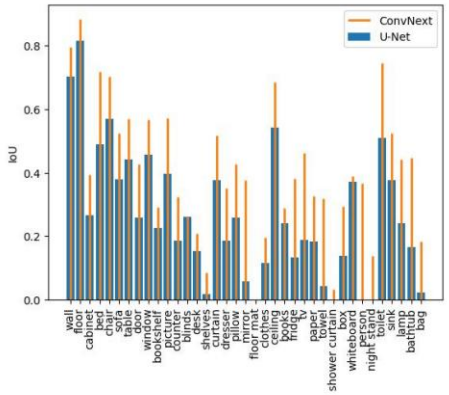


Fig. 6: IoU per class on the test set for U-Net and ConvNext

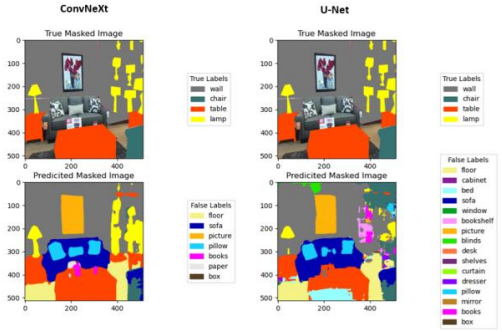


Fig. 8: Classification of general scene

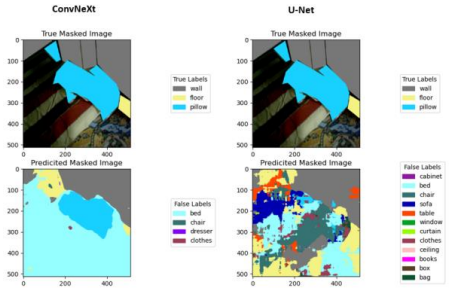


Fig. 9: Classification performance in ambiguous scene

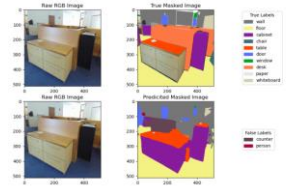


Fig. 16: Segmentation performance of ConvNeXt on a scene with difficult textures

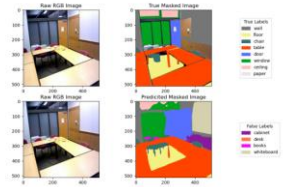


Fig. 17: Segmentation performance of ConvNeXt on a scene with ambiguous window

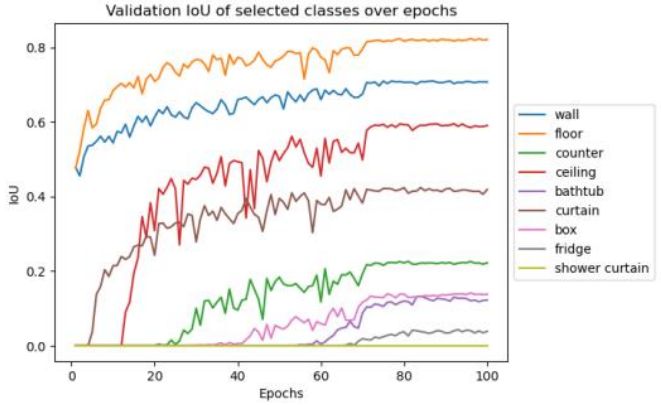


Fig. 7: U-Net validation IoU of selected classes over epochs

6. Further research

In conclusion we have presented the strength of transfer learning, the importance of data augmentation, and explored optimal hyperparameters

Conlcusions and further research

Conclusions

- *Transfer learning has a measurable and deep impact on performance in semantic segmentation*
- *Dataset augmentations allow increase the variance within the dataset allowing for longer and deeper learning*
- *We propose learning rate, and architectural set-up for a U-Net and ConvNeXt to train well for semantic image segmentations*

Further research

- *Custom pre-training of U-Net*
- *Further regularization methods*
- *Expansion of training set*
- *Further training of weighted loss models*
- *Research imbalanced based augmentation*

Thank you for listening!