FAKULTÄT FÜR MATHEMATIK, INFORMATIK UND STATISTIK
INSTITUT FÜR INFORMATIK
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
LEHRSTUHL FÜR DATENBANKSYSTEME UND DATA MINING

Lecture Notes on

Deep Learning and Artificial Intelligence

Winter Semester 2024/25

Deep Learning in Computer Vision

Lectures: Prof. Dr. Matthias Schubert

Tutorials: Maximilian Bernhard

1

Outline

- Image Classification
- Preprocessing and Image Backbones
- Methods for Object Detection
  - One-Stage Detectors
  - Two-Stage Detection
  - Transformer-Based Detectors
- Methods for Segmentation
  - U-Nets and CNN-based
  - Transformer-Based Segmentation

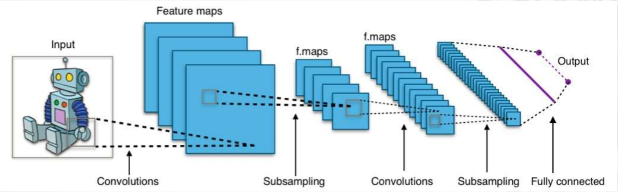
2

Imaging Architectures

- input: a height $\times$ $width \times channels$ tensor
- outputs:
  - one output (category vector) for an image:
    - scene classification (e.g. describe the main content of an image)
    - multi-class prediction (e.g. all contents on the picture)
  - on output per pixel (Segmentation)
    - semantic segmentation: all pixel describing an object of the same class have the same value (forest, water)
    - instance segmentation: all pixels belonging to the same object share a common value(pedestrian A, vehicle B)
    - Panoptic segmentation: combines instance and semantic segmentation
  - multiple sub-images: (Object Detection)
    - draws boxes around objects which might overlap
    - can be combined with instance segmentation, i.e., segment all pixels in the box representing the object

Image classification

- basic methods stack convolutional layers while shrinking height and width and increasing the channel size
- decreasing size is done with pooling layers
- a feature map yields a rescaled version of the image with various channels which potentially contains information for a pixels relying of the receptive field on the CNN encoder.
- for scene classification: the feature map is usually flattened into a vector and processed by an MLP.

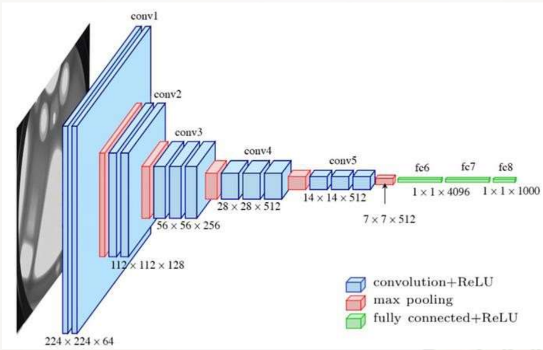


https://de.wikipedia.org/wiki/Convolutional_Neural_Network#/media/Datei:Typical_cnn.png

Image Classification

Example Architecture: VGG16

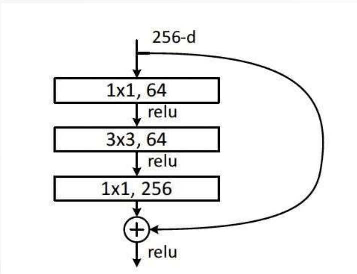
Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

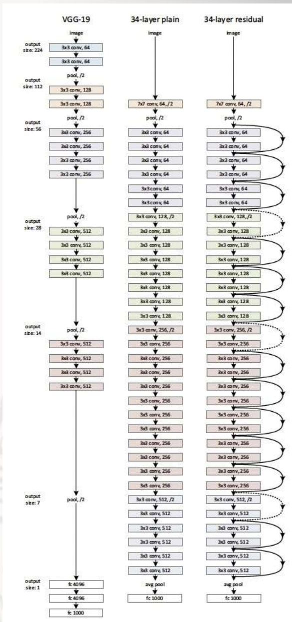


Ferguson, M., Ak, R., Lee, Y. T. T., & Law, K. H. (2017, December). Automatic localization of casting defects with convolutional neural networks. In *2017 IEEE international conference on big data (big data)* (pp. 1726-1735). IEEE.

ResNet Feature Maps

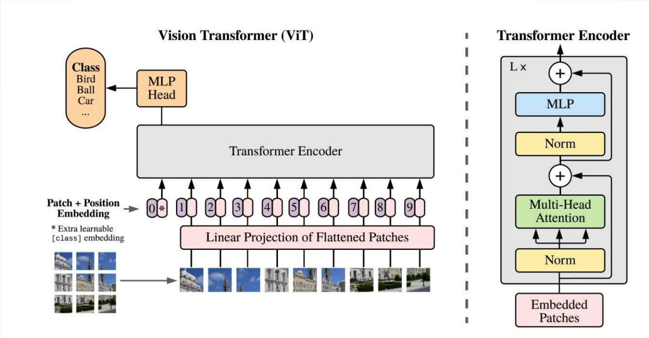
- stacking too many CNNs become inefficient and error prone (e.g. VGG-19)
- ResNet introduced residual connections to improve the gradient flow
- bottle-neck building blocks to reduce parameters:





He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Vision Transformer



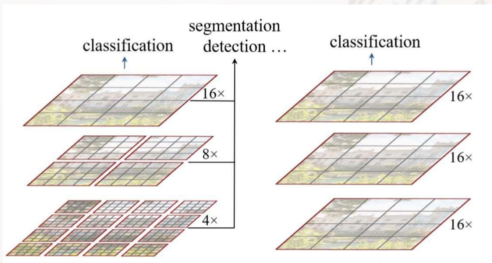
- images are partitioned into k patches which provide tokens
- positional embedding over two dimensions
- an additional class token is added and attends to all patches
- class predictions are based on the class token to avoid using all patches

Shifted Window Transformer (SWIN)

- Input of ViTs has to be of equal size
- ViT does not adapt well to differently scaled images
- ViT is quadratic concerning the image size.
  (4x4 partitioing = 16, 8x8 = 64 patches etc.)

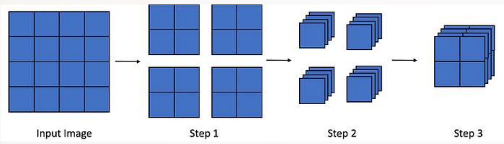
SWIN Transformer:

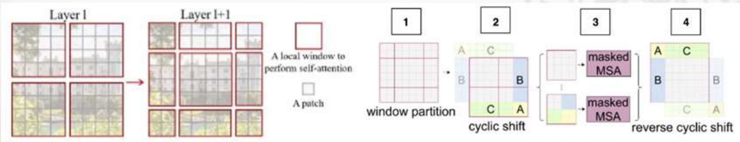
- apply local attention to patches and connect patches via window shifting
- hierarchically decrease spatial extension, and increase number of channels (c.f. CNNs)



Partitioning in SWIN

- attention is applied to each patch separately
- outputs tokens are stacked as new channels
- window shifting connects the patches and avoids quadratic global attention
- idea very similar to hierarchi CNN architectures





SWIN -Architecture

- 1 SWIN Transformer Block combines windowed and shifted window attention to cover the current resolution
- at each stage, the spatial resolution is divided by 2 in each dimension (e.g.: 256x256x3-64x64x(3x4x4),64x64xC, 32x32x2C, 16x16x4C,8x8x8C)
- normalization in multi-head self-attention includes image size:

$Attention(Q,K,V) = SoftMax\left(\frac{QK^T}{\sqrt{d}+\boldsymbol{B}}\right)V$ with $\boldsymbol{B} = M^2 \times M^2$ for M^2 patches

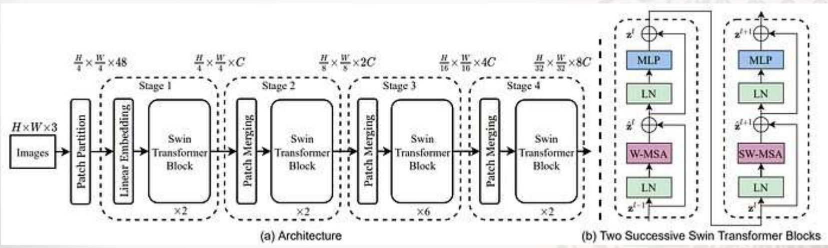
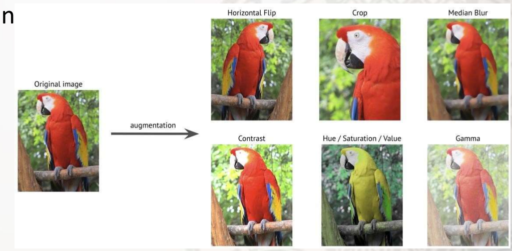


Image Augmentation

- trained models should robust against image variations which do not change the semantic.
- image augmentations apply functions on training images, which are neutral to the semantics like:
  - flips and rotation
  - color changes (hue, saturation, greyscale, etc.)
  - crop parts of an image (changes input resolution)
- in training random augmentation on all training images
- more variation and technically more training samples



https://albumentations.ai/docs/introduction/image_augmentation/

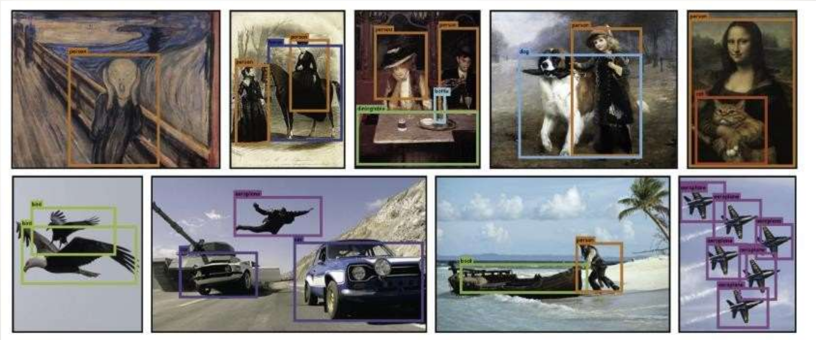
Imaging Backbones

- Image classifiers such as (ResNet, ViT, SWIN) are trained for image classification on larger image data sets
  for example: ImageNet contains 14,197,122 images being labelled with 21841 WordNetSynsets as labels
- usually, the model first learns are H x W x C ***Feature Map*** of an image which summarizes the image content
- Feature Map generation genarilizes well between tasks as the rely on low-level optical patterns.
- an image backbones consists of a pre-trained network which transforms RGB images to Feature-Maps
- there are backbones for ResNet (ResNet50/101), ViT and Swin with different sized being trained with image classification

  (next week we will see other ways of training backbones(

Object Detection

- input data: $w \times h \times c$ image
- output: $\{(y, w, c) \mid y \in \mathbb{R}^2, w \in \mathbb{R}^2, c \epsilon C\}$
  y: box centre, w: box extension, c: object class

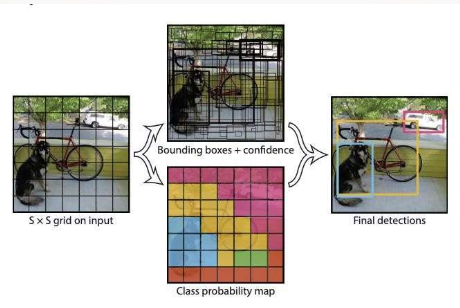


Challenges in Object Detection

- object detection evaluates image regions to determine whether they contain an object of interest. (these regions are sometimes called anchors)
- there are two types of architectures:
  - one-stage detectors learn a feature map and evaluate any entry whether it significantly overlaps an object
  - two-stage detectors use various sliding windows of varying size (anchors) to examine a large candidate list of regions, preselect some of these and scale to one size.
- on each region, a classification and box-regression head predicts the object class and centre/extension of the bounding box.
- challenges:
  - only positive ground truth is given, but no negative examples
  - when is an object correctly found? (minimum overlap with ground truth)
  - how to handle various detections partly overlapping with the same ground truth box

One-Stage Detector

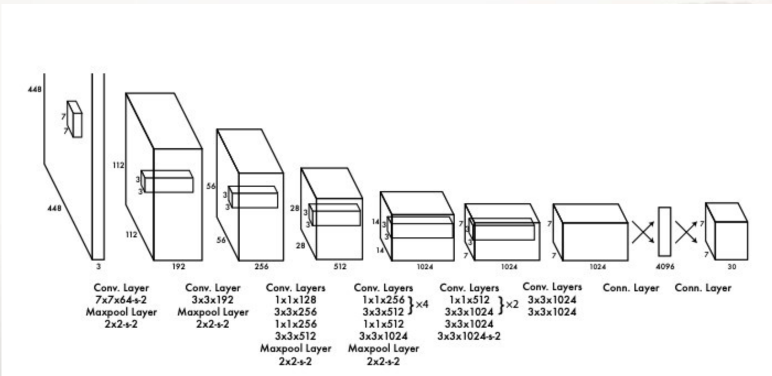
- learn a feature map and use each entry as a candidate location
- join close-by location and predict the extension of the box with a regression head for centre/extension
- most prominent YOLO (you only look once)



Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Yolo Architecture

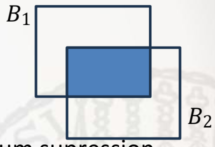
- yolo predicts 1 class label but multiple bounding boxes per output cell⇒ Select the best fitting one.
- yolo can detect at most one object per cell



Intersection over Union

- we need a metric to compute the overlap between boxes:
  Intersection over Union (IoU) computes the ratio between overlapping pixels and all pixels in all boxes.

$$IoU(B_1, B_2) = \frac{|B_1 \cap B_2|}{|B_1 \cup B_2|}$$



- is used to rank candidates during training: non-maxiumum supression
  - picks the highest confidence prediction box
  - deletes all other predictions having an IoU with more than a given threshold
  - keep doing this until the candidates are either picked or deleted
- also IoU is part of the evaluation process for object detection:
  - average precision: computes the detection precision for a class and an IoU threshold which is required to identify a true positive.
  - compute the mean over all object classes.

Two-Stage Detectors

- generate larger candidates sets and preselect them
- 1 Stage: Region of interest detection (is there an object?)
- 2 Stage: RoI evalution (predict class and bounding box)

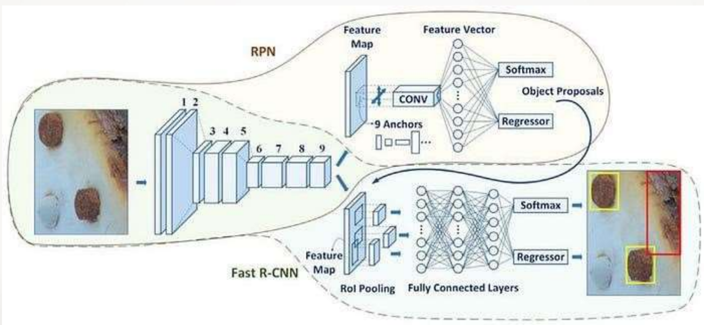
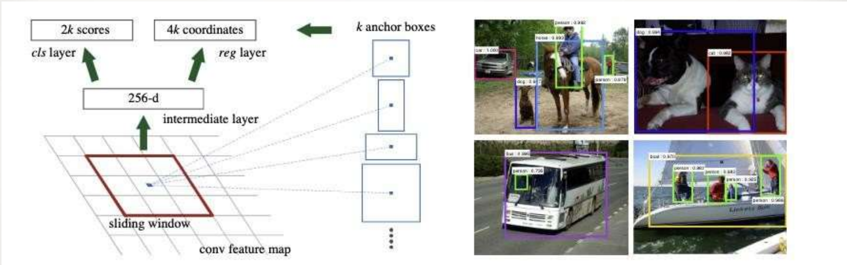


image: https://towardsdatascience.com/faster-rcnn-object-detection-f865e5ed7fc4

Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems 28 (2015).

RoI Proposal Network

- uses several sliding windows size (anchor boxes)
- for an objectness score is computed



- RoI pooling scales candidates back to a common size by summarizing pixels
- various object scales are basically handled by using differently sized anchors

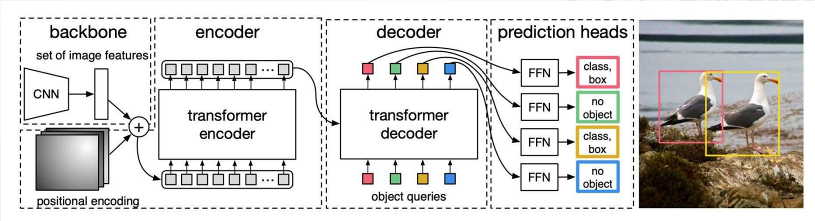
Faster RCNN vs. Yolo

- Yolo is generally faster in inference and requires less computational power in inference which makes it preferable in embedded environments like drones and robots
- Faster RCNN still yields top detection and mAP scores, but is slow due to the excessive candidate generation
- Both methods have been optimized over the years and thus, became faster and more accurate
- still there is couple of transformer based detectors (DeTR) which more and more take over.

Detection Transformers (DETR)

https://arxiv.org/abs/2005.12872

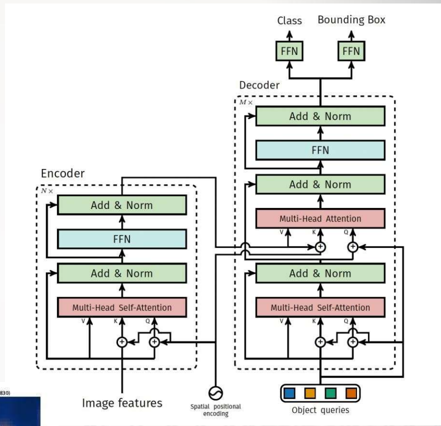
- end-to-end transformer-based object detection
- encodes the image content and decodes *k* object queries
- for each object query a class and a box (center, extension) is predicted
- if less than *k* objects present on the image, unused object queries need to be classified as ***"no object"***
- original feature map is build from CNN but could be any backbone



DETR

Architecture details:

- backbone shrinks spatial extension: take pixels as tokens for the encoder
- encoder inputs only via cross-attention
- a class head and a box head predict whether the object query corresponds to an object and where it is located
- initial object queries need to vary to find different objects



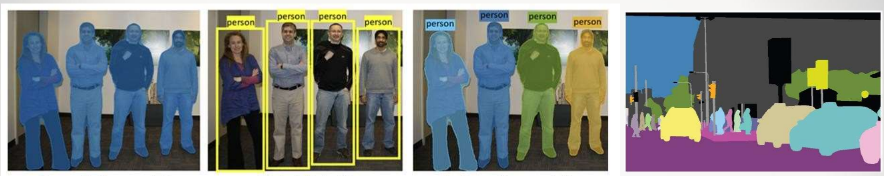
Training DETR

- to train DETR predictions must be matched with the best fitting ground truth objects (boxes and classes)
- this is achieved via computing a Hungarian Matching
  - consider a quadratic assignment matrix containing the loss between each candidate and ground truth object
  - the Hungarian Matching corresponds to the permutation of rows where the sum over the diagonal is minimal
  - computable in $O(n^3)$ with Hungarian Method

	a	b	c
x	0.6	0.3	0.6
y	0.7	0.5	0.2
z	0.8	0.1	0.5

	a	b	c
x	0.6	0.3	0.6
z	0.8	0.1	0.5
y	0.7	0.5	0.2

Image Segmentation



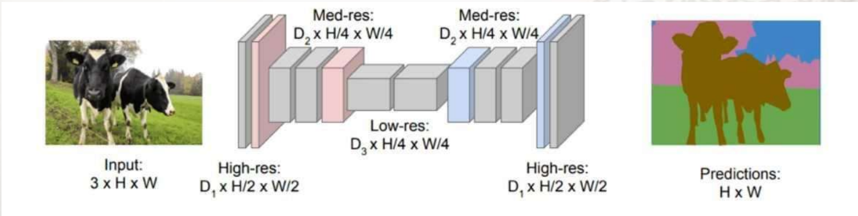
Semantic Segmentation | Object Detection | Instance Segmentation | Panoptic Segmentation

- Semantic Segmentation
  - classifies pixels into semantic classes
  - all pixels of the same class have the same label
- Instance Segmentation
  - classifies pixels into classes and instances (person_1,person_2, etc.)
  - needs to distinguish instances
  - can be done by object detection and subsequent labeling of box pixels
- Panoptic Segmentation
  - as instance segmentation but allows ambient classes like sky, gras, water which are not separable into instances

CNN-based Segmentation

- Encoder-Decoder networks:
  - Encoder generates a feature map, to increase the receptive field
  - Decoder maps the Feature Map back to the original size
  - Decoder outputs have one channel per class and softmax the channels to classify pixels

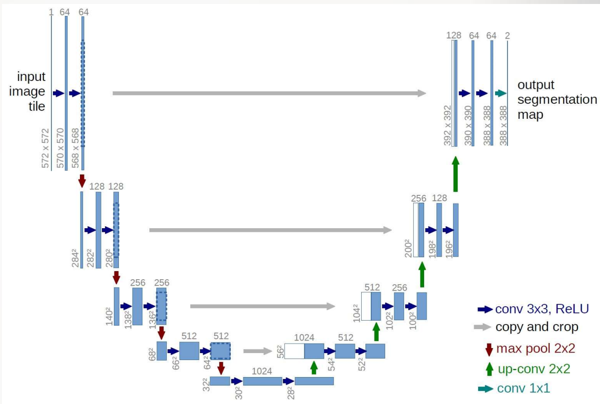
https://theaisummer.com/Semantic_Segmentation/



(image from http://cs231n.stanford.edu/)

U-Net Architecture

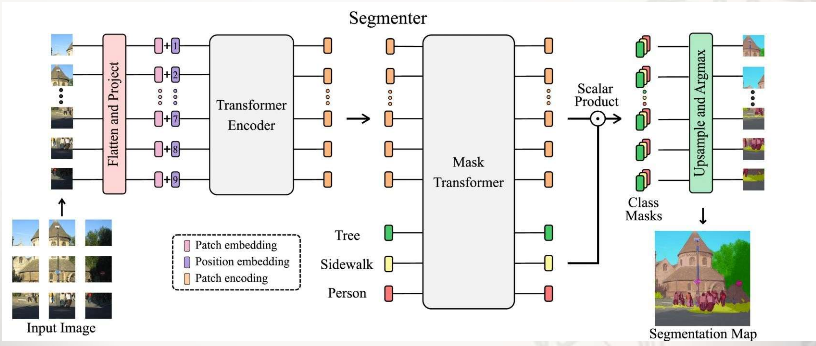
- standard CNN-Based architecture
- adds skip-connections between encoder and decoder layers of the same size
- this way details on higher resolutions don't get lost so easily
- uses up-convs to increase the output (rearrange the output channels into groups corresponding to pixels)



Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18.* Springer International Publishing, 2015

Transformer-Based Segmentation

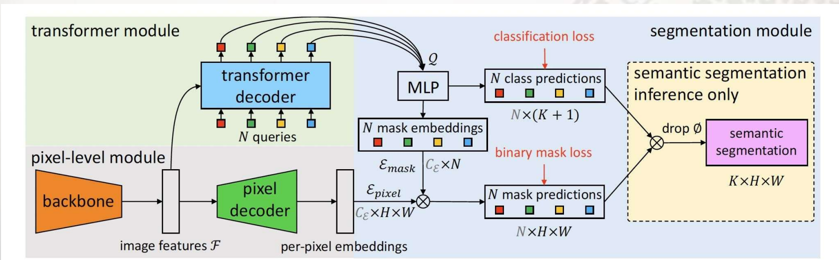
- Segmenter (https://arxiv.org/abs/2105.05633)
- Mask Transformer learns class tokens
- computing the dot product between token embedding and class tokens generate a class mask (a c-dimensional vector for each token where c is the number of classes)
- the class mask is upsampled to the original size with bilinear extrapolation



SegFormer

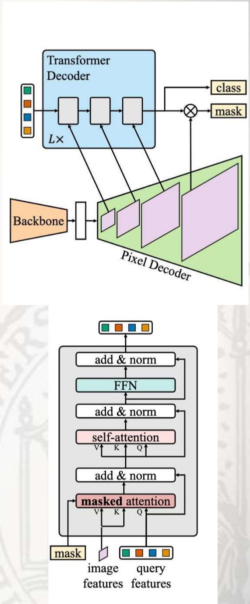
https://arxiv.org/abs/2107.06278

- ground truth with segmentation masks: $\{(c_i, m_i)$ȁ$c_i \in C, m_i \in \{0,1\}^{H\times W}\}$
- predicts N queries to generate masks and assigns masks to the classes
- loss: σ$_{j=1}^{N}\left[-\log p_{\sigma(j)}(c_j) + \mathbb{1}_{c\neq\emptyset}\mathbb{L}(m_j, m_{\sigma(j)})\right]$
- to assign candidates to ground truth pairs apply Hungarian matching
- a transformer decoder generates *N* queries which are used the generate *N* pixel masks and N corresponding class predictions
- it is possible to choose $N > |C|$ for instance segmentation



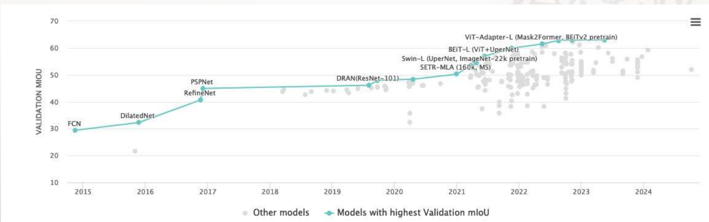
Seg2Former

- for semantic, instance and panoptic segmentation
- extends MaskFormer by:
  - interaction between Transformer and Pixel Decoder
  - multiple resolutions during decoding (better detection of small segments)
- the transformer decoder generates queries
- queries are used to generate masks
- masks are used to limit cross-attention to the area most likely corresponding to the query



Segmentation Summary

- CNN-based Segmentation is usually based on Pixel classification
- For instance segmentation, FCRNN can be used and combined with pixel classification inside the prediction box (masked-FRCNN)
- Newer methods predict segmentation masks and classes of these masks
  - all three segmentation tasks can be handled by almost identical methods
  - uses transformer decoders to generate queries describing the segments
  - queries are mapped to pixels to generate masks
  - queries are mapped to classes to classify these masks



Summary

- image classification assigns one or multiple labels to an image
- image augmentation is used to make training more robust
- backbones are pretrained networks which provide a universal mapping from an input image to a feature map
- Object Detection predicts bounding boxes and object classes
- CNN-based one-stage and two stage detectors
- Transformer-based detectors decode object queries which can be extende to boxes, instance segments and object classes
- CNN-based Segmentation like U-Net classify pixels
- SegFormer und Mask2Former predict segment masks and classes from query regions generated from a decoder