

#1. Character Region Awareness for Text Detection

by Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Clova AI Research, NAVER Corp.

[Summary]

- CRAFT(Character Region Awareness For Text detection) is designed with a convolutional neural network producing the character region score and affinity score.
 - **The region score** is used to localize individual characters in the image.
 - **The affinity score** is used to group each character into a single instance.
- To compensate for the lack of character-level annotation, this paper proposes a **weakly-supervised learning framework** that estimates character-level ground truths in existing real word-level dataset.
- Regression-based text detectors: Various text detectors **using box regression** adapted from popular object detectors
 - **TextBoxes** modified convolutional kernels and anchor boxes to effectively capture various text shapes.
 - **DMPNet** tried to further reduce the problem by incorporating quadrilateral sliding - **DSRR(Rotation-Sensitive Regression Detector)** which makes full use of rotation-invariant features by actively rotating the convolutional filters was proposed.
- Segmentation-based text detectors: This is based on works dealing with segmentation, which aims to seek text regions at the **pixel level**.
 - **SSTD(Single Shot Text Detector)** tried to benefit from both the regression and segmentation approaches by using attention mechanisms to enhance text related areas **via reducing background** interference on the feature level.
 - **TextSnake** was proposed to detect text instances by predicting **the text region and the center line together with geometry attributes**.
 - Multi-scale FCN, Holistic-prediction, Pixellink
- End-to-end text detectors train the detection and recognition modules simultaneously so as to enhance detection accuracy by leveraging the recognition result.
 - **Mask TextSpotter** took advantage of their unified model to treat the recognition task as a semantic segmentation problem.

→ This was used for text recognition instead of spotting individual characters.

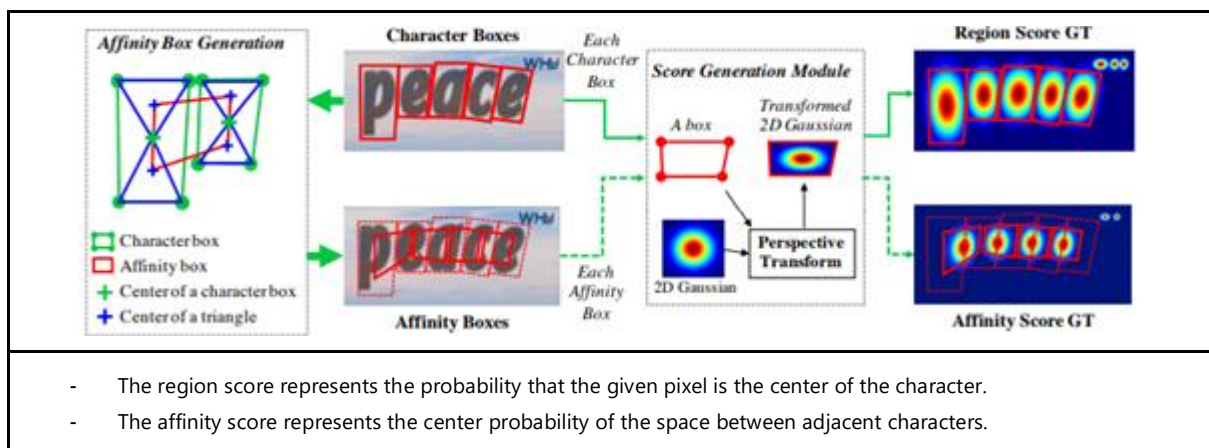
- FOTS(Fast Oriented Text Spotting), EAA

- Character-level text detector

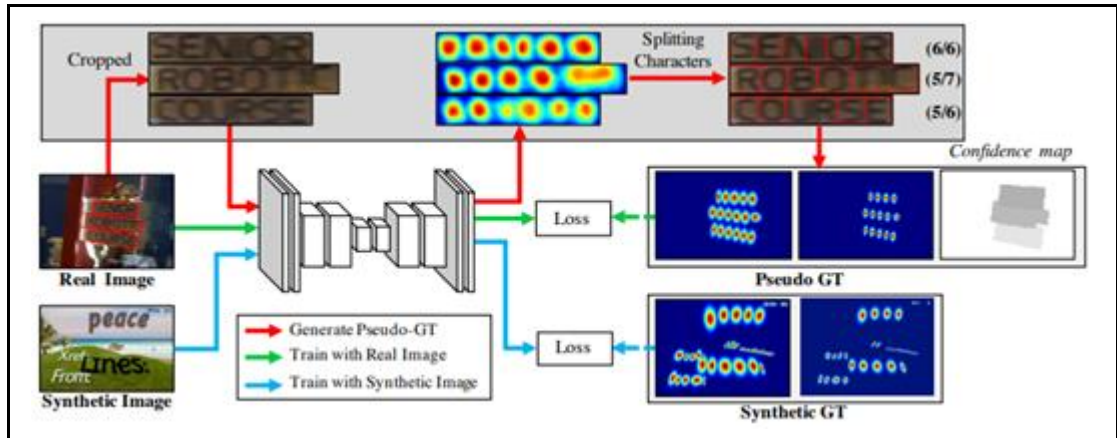
- Zhang et al. proposed a character level detector using text block candidates distilled by **MSER(Maximally stable extremal regions)**, which limits its detection robustness under certain situation, such as scenes with low contrast, curvature, and light reflection.
- Yao et al. used a prediction map of the characters **along with a map of text word regions and linking orientations that require character level annotations.**
- Instead of an explicit character level prediction, **Seglink** hunts for text grids (partial text segments) and associates these segments with an additional link prediction.

[Architecture]

: A fully convolutional network architecture based on **VGG-16** with **batch normalization** is adopted as their bone. This model has **skip connections** in the decoding part, which is **similar to U-net** in that it aggregates low-level features. The final output has two channels as score maps: the **region score** and the **affinity score**.

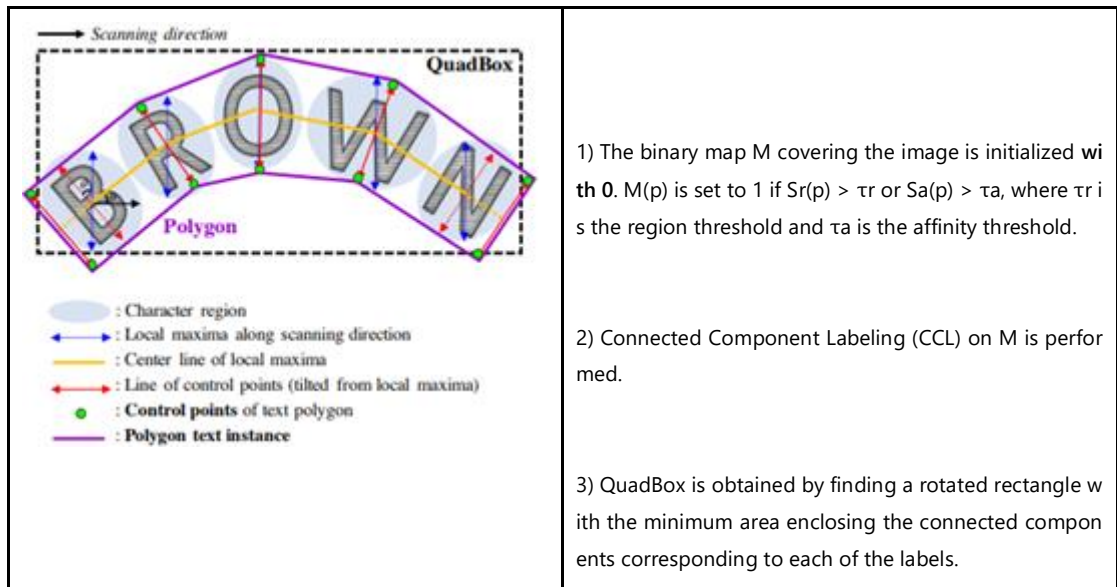


- Weakly-Supervised Learning



- 1) The word-level images are **cropped** from the original image.
- 2) The model trained up to date **predicts** the region score
- 3) The watershed algorithm is used to split the character regions, which is used to make the **character bounding boxes covering regions**.
- 4) The coordinates of the character boxes are **transformed back into the original image** coordinates using the inverse transform from the cropping step.

- Inference: In this inference stage, the final output can be delivered in various shapes, such as word boxes or character boxes, and further polygons.



- Training Strategy
 - This paper uses the **SynthText dataset** to train the network for **50k iterations**, then each benchmark dataset is adopted to fine-tune the model.
 - Some "DO NOT CARE" text regions in ICDAR 2015 and ICDAR 2017 datasets are ignored in training by **setting sconf (w) to 0**.
 - We use the **ADAM optimizer** in all training processes.

- For multi-GPU training, the training and supervision GPUs are separated, and pseudo-GTs generated by the supervision GPUs are stored in the memory.
- During fine-tuning, the SynthText dataset is also used at a rate of 1:5 to make sure that the character regions are surely separated.
- In order to filter out texture-like texts in natural scenes, On-line Hard Negative Mining is applied at a ratio of 1:3.
- Basic **data augmentation** techniques like crops, rotations, and/or color variations are applied.
- Weakly-supervised training requires two types of data, which are quadrilateral annotations for cropping word images and transcriptions for calculating word length.
→ IC13, IC15 and IC 17

#2. Multilabel Image Classification Based Fresh Concrete Mix Proportion Monitoring Using Improved Convolutional Neural Network

by Han Yang, Shuang-Jian Jiao and Feng-De Yin

① Subject: The prominent merit of the presented method lies in that it can realize real-time monitoring of fresh concrete mix proportion **only by taking pictures** which could not be achieved by previous studies and existing methods.

→ Proper and accurate mix proportion is deemed to be crucial for the concrete in service to implement its structural functions in a specific environment and structure in this paper.

② Data: Image Preprocessing, Data Augmentation and Dataset Segmentation

- The same number of **image sets obtained from 67** experiments include a total of **8340 images**.
- **Data augmentation** was carried out for preventing overfitting(rotation, horizontal shift, vertical shift, shear, zoom and horizontal flip)
- 10050 images in training set and 3350 in validation set

③ Methodology: Real-world objects always have multiple semantic meanings simultaneously.

- **Problem transformation methods** include label-based transformation and instance-based transformation, which fit data to algorithms and transform multi-label tasks into other well-established learning scenarios, especially single-label classification.

- **Algorithm adaptation methods** fit algorithms to data and adapt successful learning techniques to deal with multi-label data directly.

④ ConcMPNet was fine-tuned based on AlexNet.

- Structure about double parallel-working GPUs was ignored, convolution kernels separately deployed on two GPUs were merged and local response normalization (LRN) operation was replaced by batch normalization (BN).
- Given that training a network which has too many parameters with a relatively small dataset may lead to overfitting and further resulting in the declining generalization ability, a simpler CNN with fewer parameters was established referring to AlexNet structure.
- Input size of images, network depth, the number of layers, the number and size of filters were comprehensively considered.
- Softmax function was replaced by Sigmoid function as the activation of the output layer.
- Categorical cross entropy was replaced with binary cross entropy as a loss function to consider each output label as an independent Bernoulli distribution.
- Bayesian optimization was adopted to search optimal hyper-parameters in parameter space.
- Bayesian optimization was applied to optimize initial learning rate, batch size and epoch.

#3. Detecting Curve Text in the Wild: New Dataset and New Solution

by Yuliang Liu, Lianwen Jin, Shuaitao Zhang, Sheng Zhang College of Electronic Information Engineering South China University of Technology

① CTW1500 dataset contains 1500 images with 10,751 bounding boxes(3530 are curve bounding boxes) and at least one curve text per image.

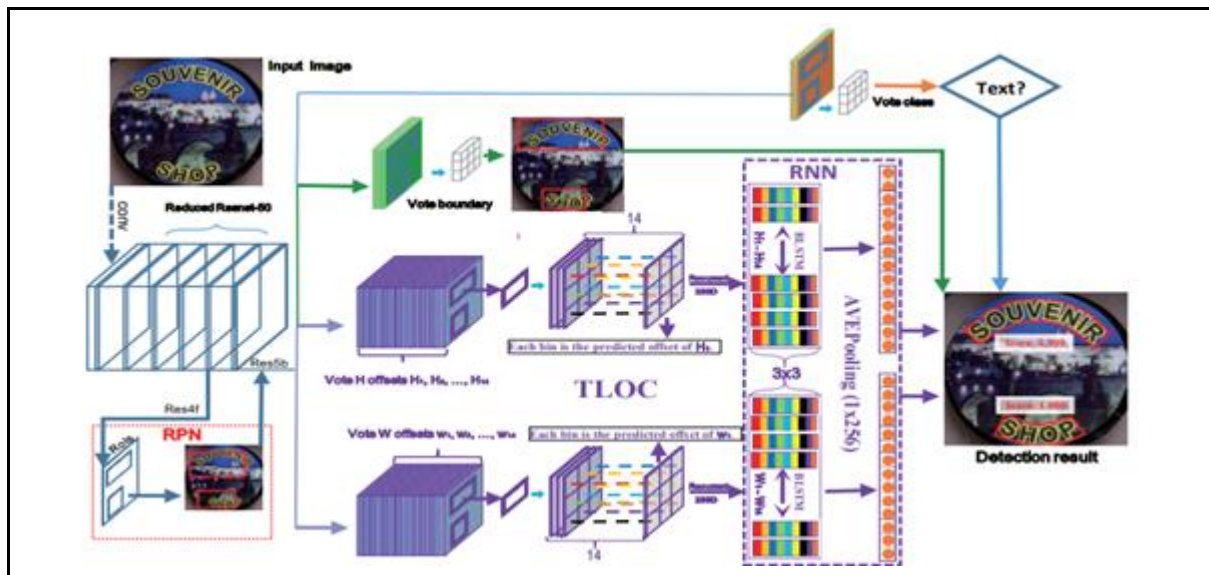
- Use the equidistant lines
- To surround the curve text, we create **ten equidistant** referenced lines to help label the **extra 10 points**(we practically find extra 10 points are sufficient to label all kinds of curve text). The reason we use the equidistant lines is to ease the labeling effort, and **reduce subjective interference**.

<https://github.com/Yuliang-Liu/Curve-Text-Detector>

- As showed Comparison of bounding boxes for localizing texts, **using curve bounding box** has three remarkable advantages

→ Avoid needless overlap, Less background noise and Avoid multiple text lines.

② CTD(curve text detector) is divided into backbone, RPN and regression module.



- **Backbone** usually adopts the popular models **pre-trained by ImageNet** and then uses the **corresponding model to finetune**, such as VGG-16, ResNet and so on.
- **RPN(Region Proposal Network)** is connected to the backbone. It generates proposal for roughly recalling text
 - They use the default rectangular anchors to roughly recall the text but we set a very loose **RPN-NMS threshold** to avoid premature suppression.
- The **Regression Module** is connected to the backbone as well and finely adjusts **the proposals to make it tighter**.
- **Recurrent TLOC(transverse and longitudinal offset connection)**
 - The transverse and longitudinal offset connection (TLOC) uses **RNN to learn the inherent connection** between locating points, making the detection more accurate and smooth.
 - To improve detection performance, **we separate transverse and longitudinal branches to predict the offsets for localizing** the text region. And each point is restricted by the last and next points and the textual region.
 - Independently predicting each offset may lead to an unsmooth text region, and somehow it may bring more false detection. Therefore, we assume **the width/height of each point has associated context information**, and use **RNN to learn their latent characteristics**.

③ Polygonal Post Processing

: Two simple but effective post-processing methods named **non-polygon suppression (NPS)** and **polygonal non-maximum suppression (PNMS)** are proposed to further intensify the generalization ability of CTD and improve the detection accuracy.


- NPS(Non-polygon suppression)
 - There is hardly any scene text coming out with intersecting sides, and these invalid polygons are nearly impossible to recognize. Therefore, we simply **suppress all these invalid polygons** and we named it a non-polygon suppression (NPS), which can slightly improve the accuracy without influencing the recall rate.
- PNMS(Polygonal non-maximum suppression)
 - Because of the particularity of the curve scene text, rectangular NMS is limited to handle dense multi-oriented text. To solve this problem, it is **proposed a locality-aware NMS and also devised a Mask-NMS** to suppress the final output results.
 - In this paper, they also improve the NMS **by computing the overlapping area between 7 polygons, named polygon non-maximum suppression (PNMS)**.

#4. What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis

by Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, Hwalsuk Lee, Clova AI Research, NAVER/LINE Corp. and Kyoto University

① Dataset Matter in STR(Scene Text Recognition)

: The dataset needs to be consistent to compare performance of models fairly. Most are synthetic data sets.

Synthetic datasets for training	
MJSynth(MJ)	SynthText(ST)
	
→ Font rendering, border and shadow rendering, background coloring, composition of font/border/background, applying projective distortion, blending with real world images and adding noise.	→ ST is a synthetically generated dataset. → ST has also been used for STR by cropping word boxes.

② Real-world datasets for evaluation

: **Seven real-world STR datasets** have been widely used for **evaluation** of a trained STR model.

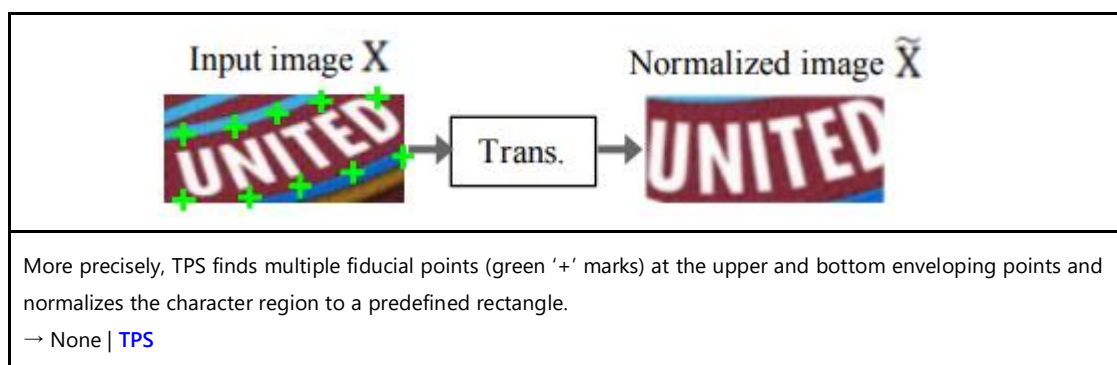
- **IIIT5K-Words (IIIT)** is the dataset crawled from Google image searches such as billboard, Sign board, house numbers and so on.(Training 2000/ Evaluation 3000)
- **Street View Text (SVT)** contains outdoor street images, which is noisy, blurry, or low-resolution collected from Google Street View.(Training 257/Evaluation 647)
- **ICDAR2003 (IC03)** was created for reading camera captured scene texts.(Training 1156/ Evaluation 860)

- **ICDAR2013 (IC13)** inherits most of IC03's images.(Training 848/Evaluation 1015)
- **ICDAR2015 (IC15)** is the dataset captured by Google Glasses while under the natural movements of the wearer. (Training 4468/ Evaluation 1811 or 2077)
- **SVT Perspective (SP)** is collected from Google Street View(Evaluation 645)
- **CUTE80 (CT)** is collected from natural scenes(Evaluation 288)

③ STR Framework Analysis

: **Transformation Stage** normalizes the input text image using the Spatial Transformer Network to ease downstream stages.

- To reduce this burden, which needs to learn an invariant representation with respect to such geometry, **thin-plate spline(TPS) transformation** has been applied with its flexibility to diverse aspect ratio of text line.

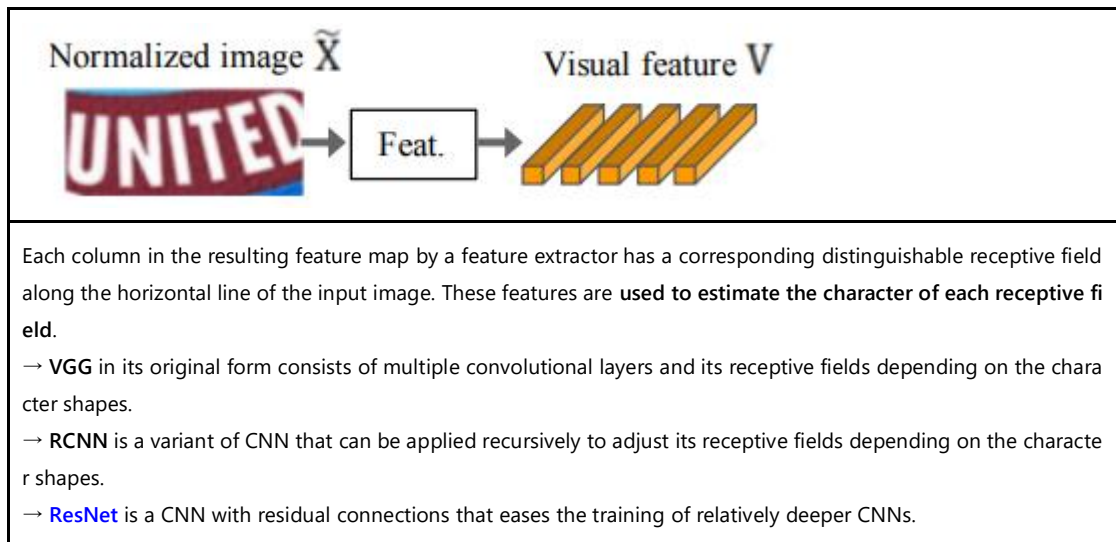


⇒ Appendix D delivers its mathematical background and the implementation.

- ① Localization network: finding a text boundary
- ② Grid generator: linking the location of the pixels in the boundary to those of the normalized images
- ③ Image sampler: generating a normalized image by using the values of pixels and the linking information
- ④ TPS-Implementation: TPS requires the localization network calculating fiducial points of an input image
 - Final output is 2F dimensional vector which corresponds to the value of x,y-coordinates of F fiducial points on input image.

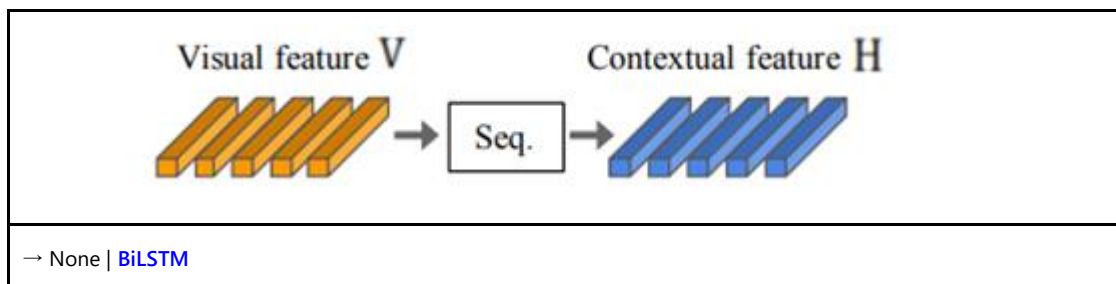
④ **Feature extraction** maps the to a representation that **focuses on the attributes relevant for character recognition**, while **suppressing** irrelevant features such as font, color, size, and background.

- A CNN abstract an input image and outputs a visual feature map.



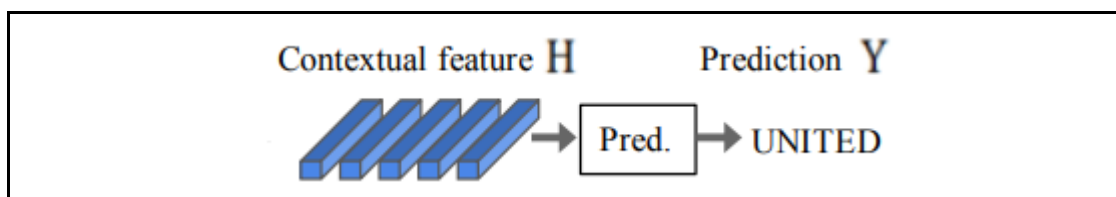
⑤ **Sequence modeling captures the contextual information** within a sequence of characters for the next stage to predict each character more robustly, rather than doing it independently.

- The extracted features from Feat. stages are reshaped to be a **sequence of feature V**. Each column in a feature map is used as a frame of the sequence but, this sequence may suffer the lack of contextual information. Therefore, some previous works use **Bidirectional LSTM(BiLSTM)** to make a better sequence $H = \text{Seq}(V)$ after the feature extraction stage.



⑥ **Prediction estimates the output character sequence** from the identified features of an image.

- **CTC(Connectionist temporal classification)** allows for the prediction of a non-fixed number of a sequence even though a fixed number of the features are given. The key methods for CTC are to predict a character at each column and to modify the full character sequence into a non-fixed stream of characters by deleting repeated characters and blanks.
- **Attn(Attention-based sequence prediction)** automatically captures the information flow within the input sequence to predict the output sequence.



For example, you make the character label set C which includes 36 alphanumeric characters.	
For the CTC,	For the Attn,
an additional blank token is added to the label set due to the characteristics of the CTC.	an additional end of sentence(EOS) token is added to the label set due to the characteristics of the Attn.

→ need to understand...

⇒ The **Blue Bold**s mean compositions of the best combination for highest accuracy.

⑦ Experiment and Analysis









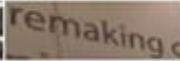



- Implementation detail(for remind)
 - **AdaDelta optimizer**, whose **decay rate** is set to $\rho = 0.95$.
 - All parameters are initialized with **He's method**.
 - This study use the union of the training sets IC13, IC15, IIIT, and SVT **as the validation data**
 - **The validated the model after every 2000 training steps** to select the model with the highest accuracy
 - **For accuracy**, we measure the success rate of word predictions per image on the 9 real-world evaluation datasets involving all subsets of the benchmarks, as well as a **unified evaluation dataset** (8,539 images in total); 3,000 from IIIT, 647 from SVT, 867 from IC03, 1015 from IC13, 2,077 from IC15, 645 from SP, and 288 from CT.
 - **Only alphabets and digits** are evaluated.
 - **For speed assessment**, we measure the **per-image average clock time** (in millisecond) for recognizing the given texts under the same compute environment.
 - **For memory assessment**, we count the number of trainable floating point parameters in the entire STR pipeline.

⑧ Module analysis

- When comparing accuracy improvement versus time usage, a sequence of ResNet, BiLSTM, TP S, and Attn is the most efficient upgrade order of the modules from a base combination of None-VGG-None-CTC. This order is the same order of combinations for the accuracy-time frontiers ($T1 \rightarrow T5$).
- On the other hand, an accuracy-memory perspective finds RCNN, Attn, TPS, BiLSTM and ResNet as the most efficient upgrading order for the modules, like the order of the accuracy-memory frontiers ($P1 \rightarrow P5$).

- Qualitative analysis

Figure 7 shows samples that are only correctly recognized when certain modules are upgraded (e.g. from VGG to ResNet backbone).

Trans. (None→TPS)			
Feat. (VGG→ResNet)			
Seq. (None→BiLSTM)			
Pred. (CTC→Attn)			

- **TPS transformation normalizes** curved and perspective texts into a standardized view. Predicted results show dramatic improvements especially for "POLICE" in a circled brand logo and "AIRWAYS" in a perspective view of a storefront sign.

- Advanced feature extractor, **ResNet**, results in better representation power, improving on cases with heavy background clutter "YMCA", "CITYARTS") and unseen fonts ("NEUMOS").

- **BiLSTM** leads to better context modeling by adjusting the receptive field; it can ignore unrelated cropped characters ("l" at the end of "EXIT", "C" at the end of "G20").

- **Attention** including implicit character level language modeling finds missing or occluded characters, such as "a" in "Hard", "t" in "to", and "S" in "HOUSE".

⑨ Failure case analysis

→ Calligraphic font, Vertical texts, Special characters, Heavy occlusions, Low resolution and Label noise

⑩ Appendix E: STR Framework - full experiment result

- VGG gives the lowest accuracy on average for the lowest amount of inference time required.(time ↓ , accuracy ↓)
 - RCNN achieves higher accuracy over VGG by taking the longest time for inferencing and the lowest memory usage out of the three.(time ↑ ↑ , memory ↓ , accuracy ↑)
 - ResNet, exhibits the highest accuracy at the cost of using significantly more memory than the other modules.(memory ↑ ↑ , accuracy ↑ ↑)
- If the system to be implemented is constrained by memory, RCNN offers the best trade-off .
- If the system requires high accuracy, ResNet should be used.

<https://ropiens.tistory.com/23>

[Additional]

① **AdaDelta(Adaptive Delta)** Optimizer is Gradient descent method combining Adagrad, RMSprop, and Momentum. It is proposed to compensate for the shortcomings of AdaGrad.

- Change the sum of the squares of the gradient of all steps, which is an Adagrad characteristic, to the sum of the window size by setting the window size. After that, apply the moving average, which is the same as RMSprop.
- In original paper, it is stated that the unit of the weight and the amount of weight change must be the same, and it is explained that SGD, Momentum, and Adagrad do not have an exact unit because the update includes the ratio of the gradient amount, and thus there is no step in updating.
- This is applied to the Hessian inverse matrix, adding the advantage of the inverse matrix.

<https://twinw.tistory.com/247>

<http://shuuki4.github.io/deep%20learning/2016/05/20/Gradient-Descent-Algorithm-Overview.html>

<http://incredible.ai/artificial-intelligence/2017/04/10/Optimizer-Adadelta/>

② **Gradient clipping** is a method of limiting the size of the norm of the neural network parameter theta. The direction of the gradient vector is maintained, but the size can be reduced to the extent that training is not damaged.

- If you are using an optimizer with a dynamic learning rate like Adam, it is not necessarily used, but it is sometimes applied as a safety device.

<https://kh-kim.gitbook.io/natural-language-processing-with-pytorch/00-cover-6/05-gradient-clipping>

<https://dhhwang89.tistory.com/90>

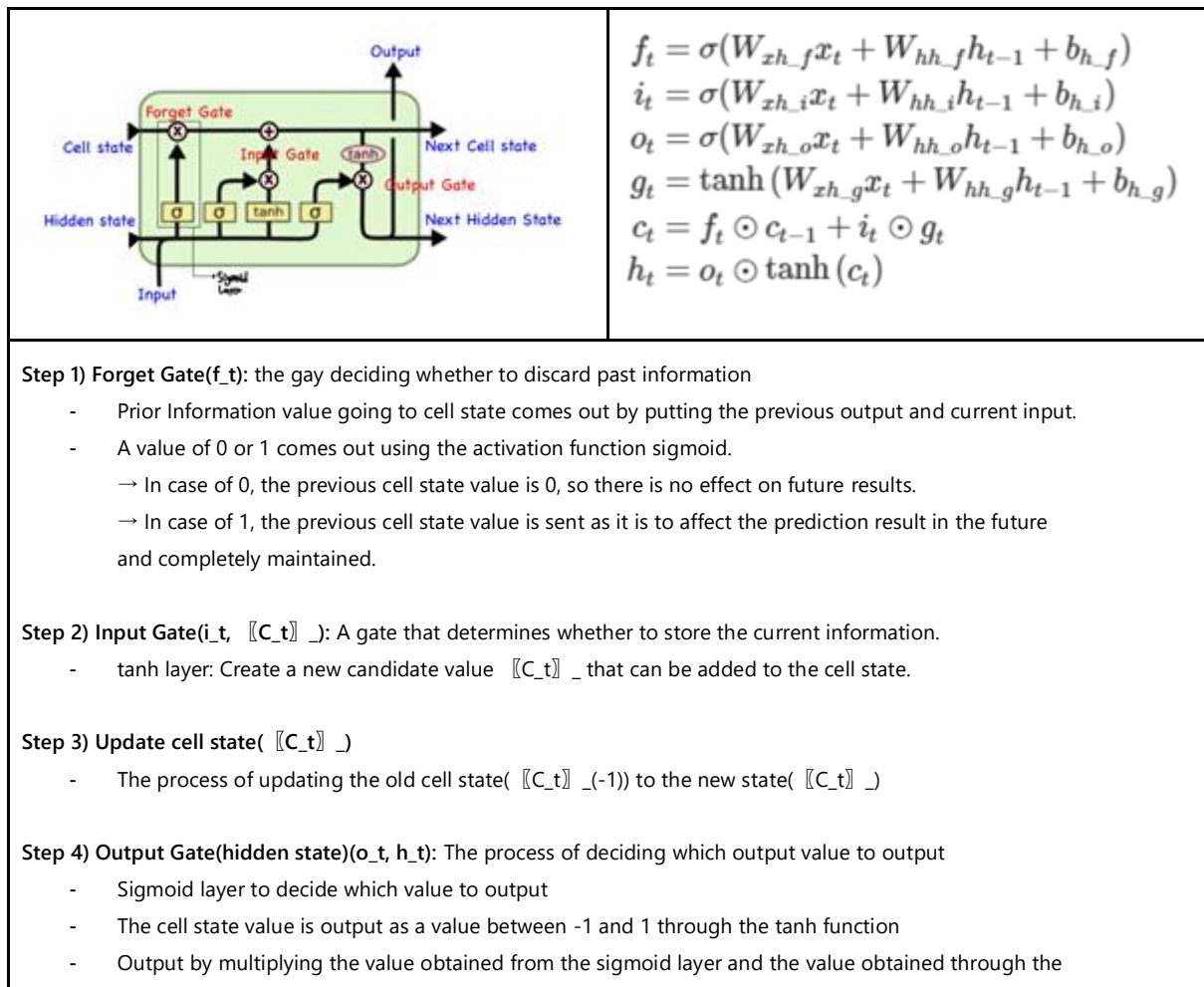
③ **NSML(NAVER Smart Machine Learning)** platform is a machine learning as a service (MLaaS), a cloud platform, designed to eliminate unnecessary work in research and make efficient use of GPU resources by NAVER.

https://n-clair.github.io/ai-docs/_build/html/en_US/index.html

④ **LSTM(Long Short Term Memory)** is one of the main models of RNN, it can solve the long-term dependency problem. Since it is a network structure that can accept inputs and outputs regardless of the length of the sequence, it has a great advantage that the structure can be flexibly created as needed.

※ The difficulty with long-term dependencies arise from exponentially smaller weights given to long-term interactions.

LSTM Structure



<https://wegonnamakeit.tistory.com/7>

<https://ratsgo.github.io/natural%20language%20processing/2017/03/09/rnnlstm/>

<https://m.blog.naver.com/PostView.nhn?blogId=magnking&logNo=221311273459&proxyReferer=https:%2F%2Fwww.google.c>

[om%2F](#)

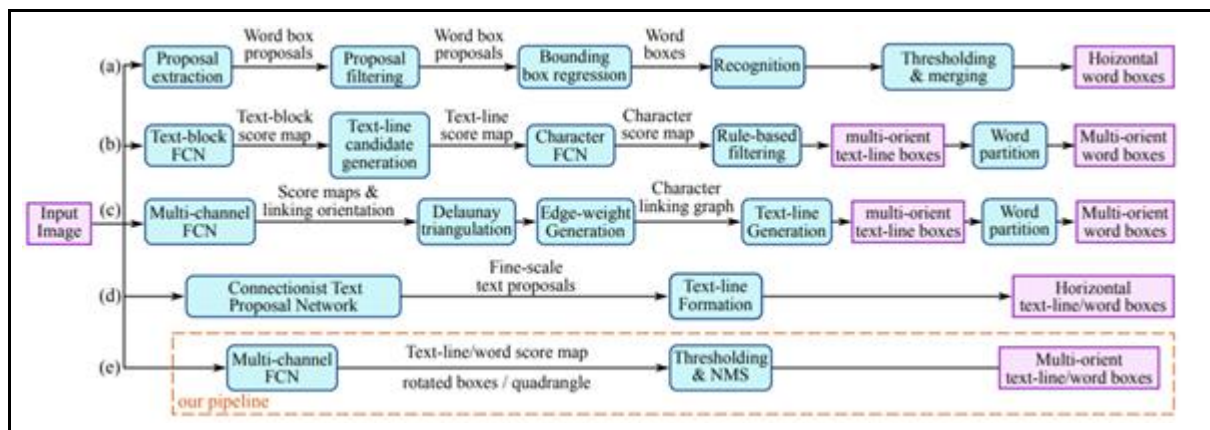
<https://wikidocs.net/22888>

#5. EAST_An Efficient and Accurate Scene Text Detector

by Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang Megvii Technology Inc., Beijing, China

① Comparison of pipelines of scene text detection

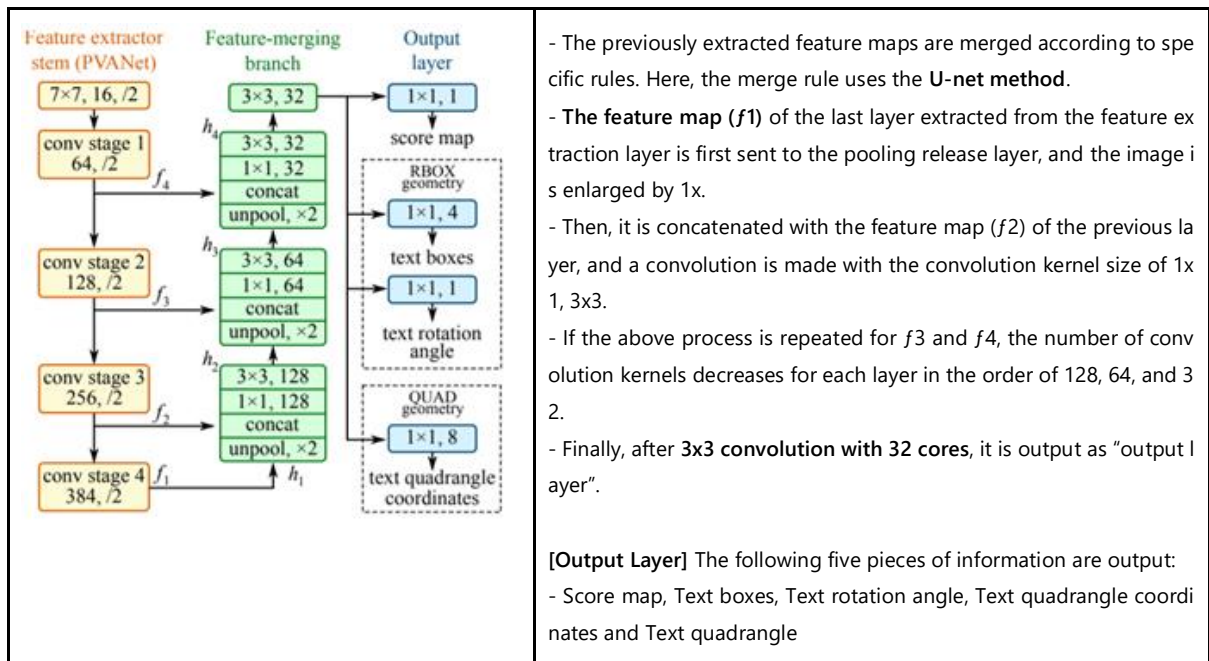
: EAST(An Efficient and Accurate Scene Text Detector) is a text detection model that simplifies the intermediate process step, realizes end-to-end text detection, and further improves detection accuracy and speed.



- In the image above, (a), (b), (c) are typical text detection processes. Common detection involves **extraction of candidate boxes** and **filtering of candidate boxes**.
- (d) is a CTPN(Connectionist Text Proposal Network) model. The detection process is similar to the EAST model in (e), but **only supports horizontal text detection** and the applied image is not as effective as the EAST model.
- (e) is the EAST model detection process.
 - As can be seen from the figure above, the whole process is simplified **to only the FCN stage and the NMS stage**, and the intermediate process is greatly reduced.
 - The output result **supports multi-angle detection of text** and can be applied to various application scenarios.

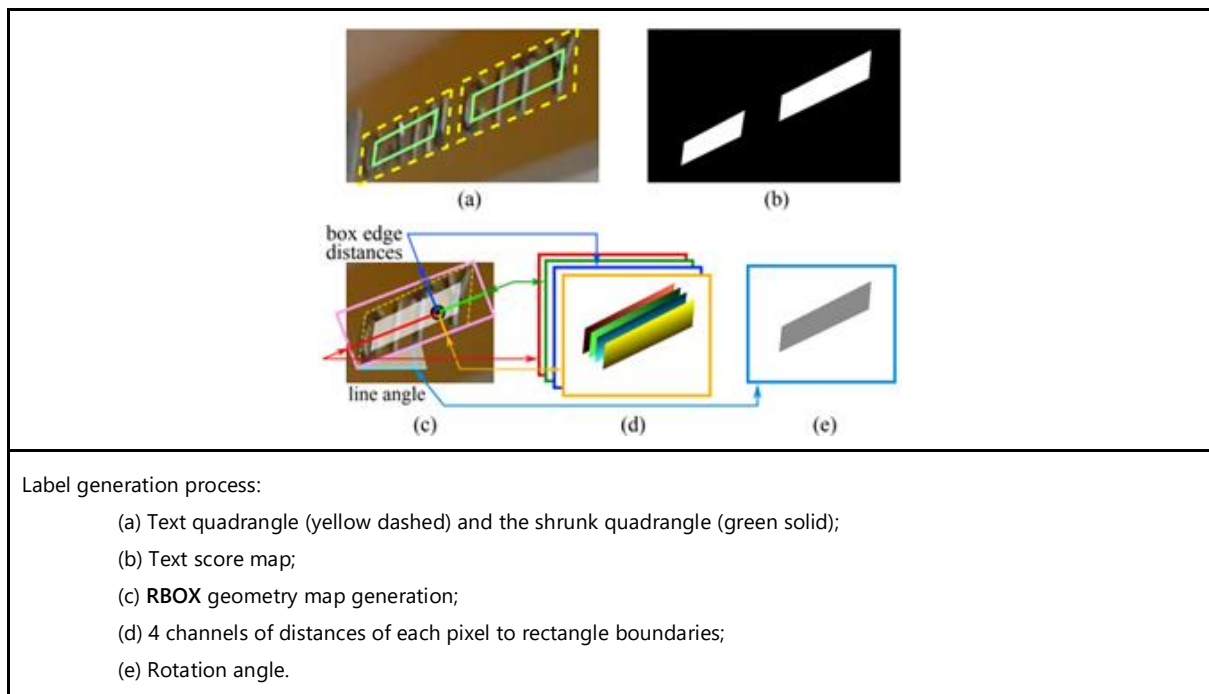
② EAST model network structure

	<p>[Feature Extractor Stem(PVANet)]</p> <ul style="list-style-type: none"> - Based on PVANet as the backbone of the network structure, feature maps are extracted from the convolution layers of stage 1, stage 2, stage 3 and stage 4, respectively. - Four levels of feature maps, denoted as s_1, s_2, s_3, s_4, are extracted from the stem, whose sizes are $1/32, 1/16, 1/8$ and $1/4$ of the input image, respectively. - The model can be decomposed into three parts: feature extractor 'stem', feature-merging branch and output layer. - The stem can be a convolutional network pre-trained on ImageNet dataset, with interleaving convolution and pooling layers. - In this way, text lines can be detected by extracting feature maps of various scales. <p>[Feature-merging Branch]</p>
--	---



PVANet <https://arxiv.org/abs/1608.08021>

+) Output geometry design



- **AABB(Axis-Aligned Bounding Box)**: Distance information from each point in an arbitrary shape rectangle to the four sides of the rectangle.
- **RBOX(Rotated Box)**: AABB information and oblique angle information of rectangle.
- **QUAD**: The difference in coordinates from each point in the arbitrary shape rectangle to the vertices of the 4 arbitrary shape rectangles.

③ Loss Functions

$L = L_s + \lambda_g L_g$	<p>The total loss function consists of score map loss (L_s) and geometry loss (L_g) as shown in the above equation.</p> <p>Lambda is a hyper-parameter that adjusts the weight for each loss. In this study, it was set to 1.</p>
---------------------------	---

- **Score loss** is a loss to accurately estimate the score map composed of **positive-negative**.
 - In most object detection tasks, the 'imbalanced' object distribution problem is solved through '**balance**' **positive-negative sampling and hard negative**.
 - This process not only includes a non-differentiable process, but also increases processing steps, which can lead to problems such as error-propagation. To compensate for these shortcomings, in this study, **positive-negative sampling balance** was applied to score map loss. A balanced cross entropy loss that can be set automatically is applied.
 - **Balanced cross entropy loss** reflects negative samples as well as positive samples to the loss, and in this case, **the weight of the cross entropy item is set in consideration of the ratio of positive-negative samples**.
 - **Geometry loss** is a loss set in **RBOX** and **QUAD** information to accurately estimate the **four vertices of an arbitrary rectangle**.
 - When the L1 or L2 distance measure for the position of a vertex is applied, the loss of a sample with a large character size is relatively larger than that of a sample with a small character size, so small characters may not be recognized well. Therefore, it is necessary to set the geometry loss **with scale-invariant characteristics**.
 - To this end, in this study, **IoU (Intersection over Union)** and **RBOX angle-based** loss functions were defined for AABB estimation.
- +) **IoU (Intersection over Union)** is an evaluation metric used to measure the accuracy of an object detector on a particular dataset.

<https://ballentain.tistory.com/12>

④ Locality-Aware NMS

- **NMS(Non Maximum Suppression) Algorithm** is a concept used in overall image processing. It maintains only the maximum value between overlapping areas (or pixels) and eliminates the non-maximum one.
 - In the case of the EAST algorithm, the probability of a letter is predicted in units of pixels, but there is a problem of predicting the same character area by overlapping. At this time, the problem is solved by applying the NMS algorithm **to remove only the char**

acter area with the highest probability among the plurality of character areas overlapping a certain part or more.

- First, all the output box sets are combined with the corresponding threshold value (if it is greater than the threshold value, it is combined; if it is less than the threshold value, it is not combined), and the reliability score is used as a weight combination to obtain the combined box set. Perform standard NMS operations on the merged RBOX collection.

⑤ Experiments

- Trained end-to-end using ADAM optimizer.
- To speed up learning, we uniformly sample 512x512 crops from images to form a minibatch of size 24.

#6. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

by Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas and Stanford University

[Summary]

① **PointNet** is a unified architecture that **directly takes point clouds as input and outputs either class labels for the entire input or per point segment/part labels for each point of the input.**

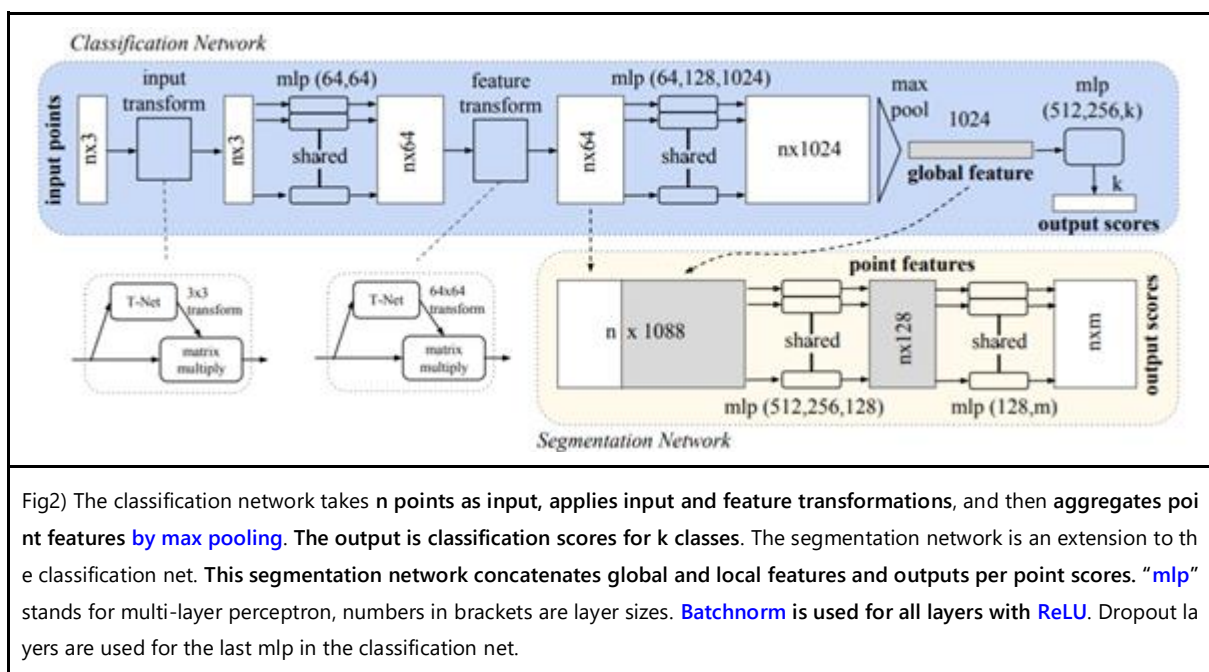


Fig2) The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net. This segmentation network concatenates global and local features and outputs per point scores. "mlp" stands for multi-layer perceptron, numbers in brackets are layer sizes. Batchnorm is used for all layers with ReLU. Dropout layers are used for the last mlp in the classification net.

② Network Architecture and Training Details

[Point Classification Network]

: More details on the joint alignment/transformation network and training parameters.

- **First Transformation network** is a mini-PointNet that takes raw point clouds as input and regresses to a 3×3 matrix.
 - It's composed of a shared MLP(64, 128, 1024) network (with layer output sizes 64, 128, 1024) on each point, a max pooling across points and two fully connected layers with output sizes 512, 256.
 - The output matrix is initialized as an identity matrix.
 - All layers, except the last one, include ReLU and batch normalization.
- **The second transformation network** has the same architecture as the first one except that the output is a 64×64 matrix.
 - The matrix is also initialized as an identity.
 - A regularization loss (with weight 0.001) is added to the softmax classification loss to make the matrix close to orthogonal.

+) **Using dropout with keep ratio 0.7 on the last fully connected layer**, whose output dimension 256, before class score prediction. The decay rate for batch normalization starts with 0.5 and is gradually increased to 0.99. Using **Adam optimizer** with initial learning rate 0.001, momentum 0.9 and batch size 32. The learning rate is divided by 2 every 20 epochs.

[PointNet segmentation]

: Network The segmentation network is an extension to the classification PointNet.

- Local point features (the output after the second transformation network) and global feature (output of the max pooling) are concatenated for each point.
- No dropout is used for segmentation network.
- Training parameters are the same as the classification network.
- We add a one-hot vector indicating the class of the input and concatenate it with the max pooling layer's output.

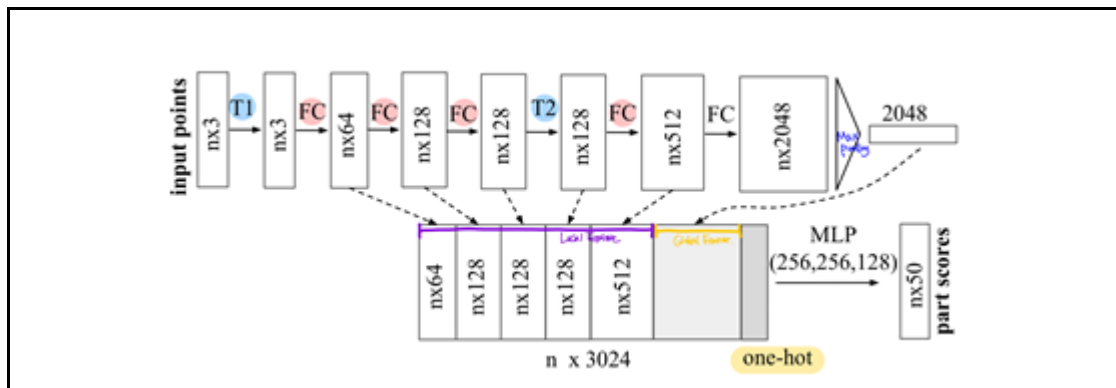
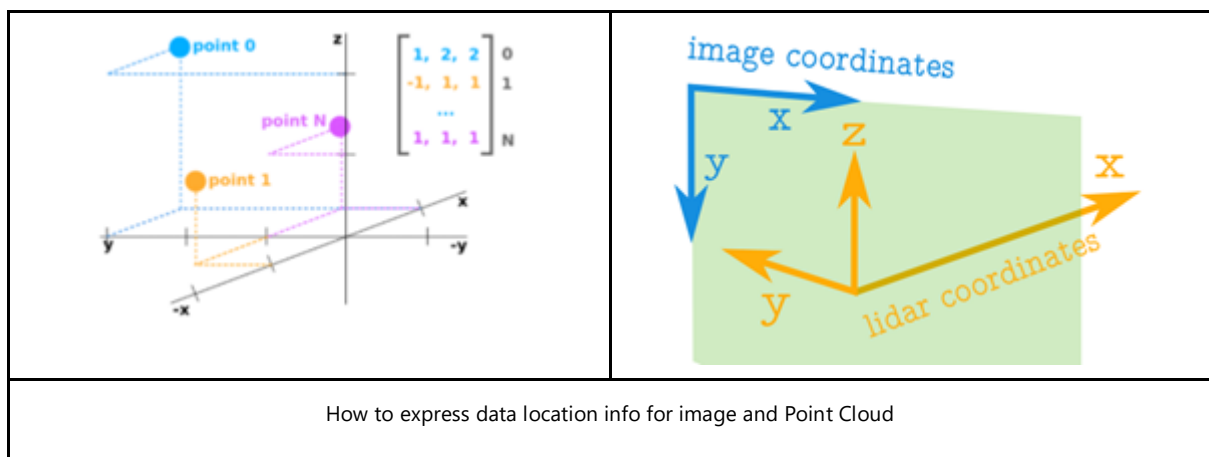


Fig 9) Network architecture for part segmentation

- T1 and T2 are alignment/transformation networks for input points and features.
- FC are fully connected layers operating on each point.
- MLP is a multi-layer perceptron on each point.
- One-hot is a vector of size 16 indicating the category of the input shape.

[Additional] Based on Ms.Hwang's summary

① **Point Cloud**: Usually collected by Lidar sensor, RGB-D sensor, etc. These sensors send **light/signal** of an object and record the return time, **calculate distance information for each light/signal**, and generate one point.



+) **Point Cloud Filtering**: The ROI can be gotten by setting the (x, y, z) range in 3-D in the point cloud such as a region of interest in an image.

Point Cloud	https://info.vercator.com/blog/what-are-point-clouds-5-easy-facts-that-explain-point-clouds
Additional Point Cloud	https://blog.naver.com/nswve/222170343140
Permutation invariant	https://blog.naver.com/qbxlvnf11/221659870504
Voxelization	https://pcl.gitbook.io/tutorial/part-1/part01-chapter02
Rendering	https://parksh86.tistory.com/168
Mesh	https://hellowoori.tistory.com/30

