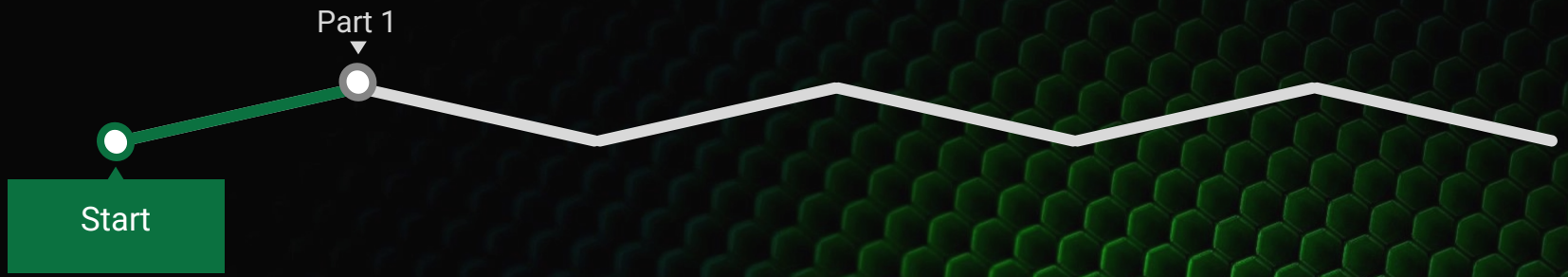
Indian Institute of Space Science and Technology, Thiruvananthapuram

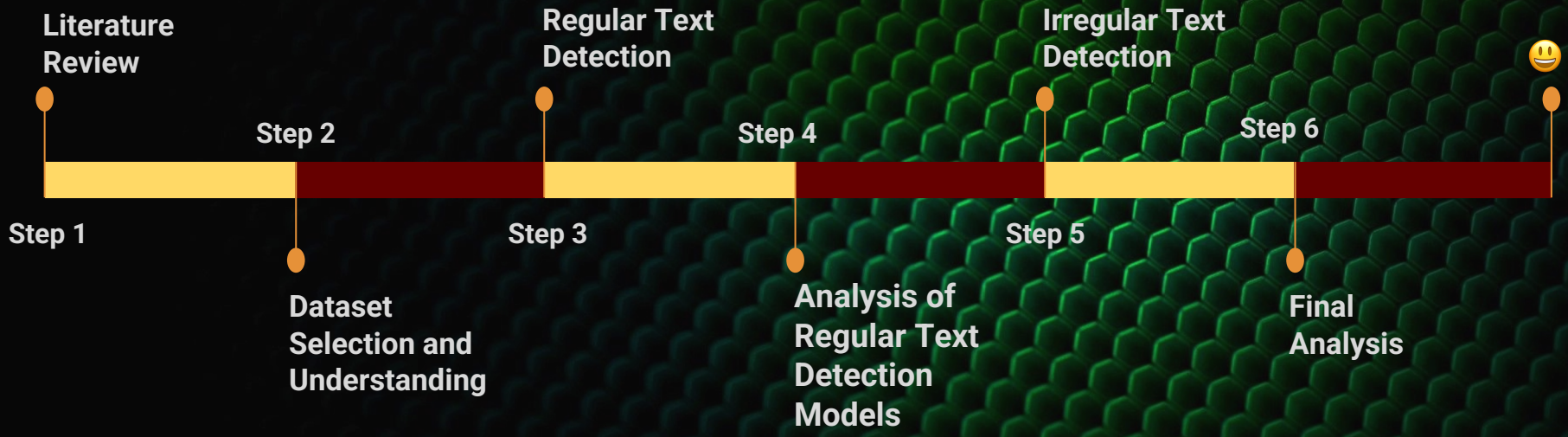
Scene Text Recognition

AV489 Machine Learning for Signal Processing
Course Project



Introduction and Overview

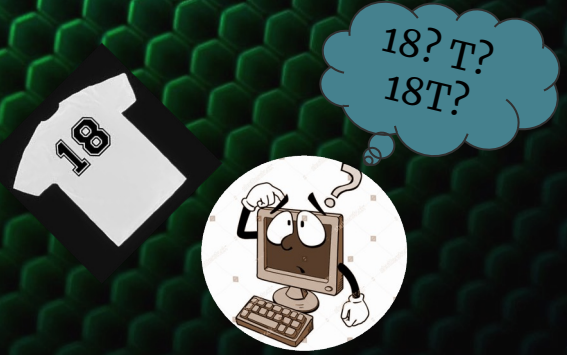
Our Workflow



Objective

- Image-based sequence recognition
  - Longstanding research topic
  - Includes Text, Numbers, special characters, patterns etc.

- Our problem Statement : **Scene Text Recognition**
  - Most important and challenging task in image-based sequence recognition.
  - Includes Text and Numbers
  - Two Type :
    - Regular Text
    - Irregular Text



Regular Text vs Irregular Text

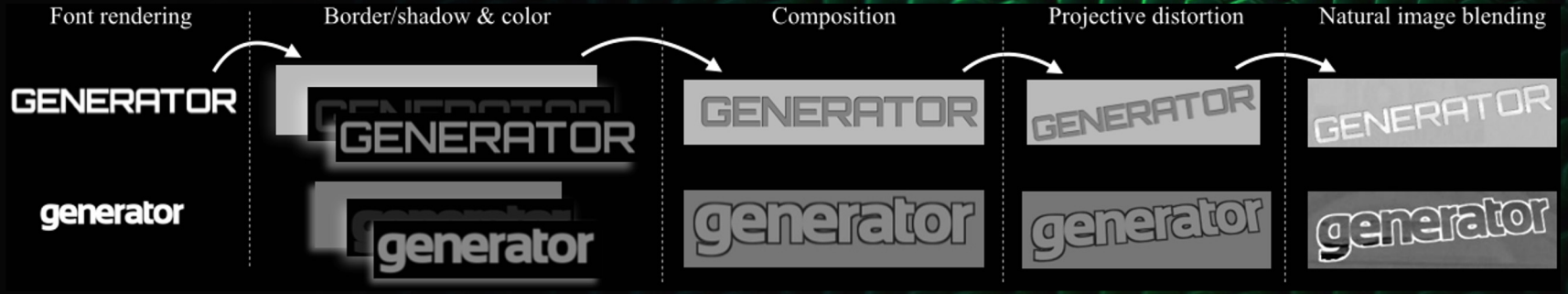
- Regular form
- From Left to Right with 0 inclination

- Irregular form
- Curved or Inclined images
- Try to make it regular through transformation network before using

Training Datasets

Synth90k

- Synthetically generated
- 7.2 million training images
- Total 9 million images covering 90k English words



Testing Datasets

IIIT5k

- 5000 cropped images developed from google image search
- Mostly Regular text

Street View Text (SVT)

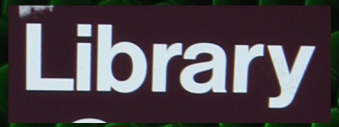
- Harvested from Google Street View.
- Contains distortions and blur
- 647 word uncropped images
- Mostly Irregular text

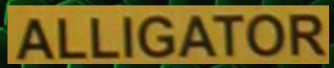
Street View Text Perspective (SVTP)

- 645 word cropped images from SVT
- Has perspective distortions due to the camera viewpoint angle.

Testing Datasets

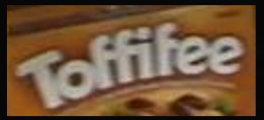
ICDAR13





- From ICDAR 2013 Robust Reading Competition
- Contains 1,095 word images including ICDAR03.
- Mostly Regular Text

ICDAR15



- Contains 2,077 word images including ICDAR13.
- Cropped from video frames
- Differs from other ICDAR dataset for real-Life factors like:
  - Occlusions,Motion blur, Noise, and Illumination factors
- Contains Both Regular as well as Irregular

Testing Datasets

Curved Text (CUTE80)

- Mainly contains 288 high resolution word images curved and/or oriented text instances.
- Dataset was originally proposed for text detection but later used for recognition.

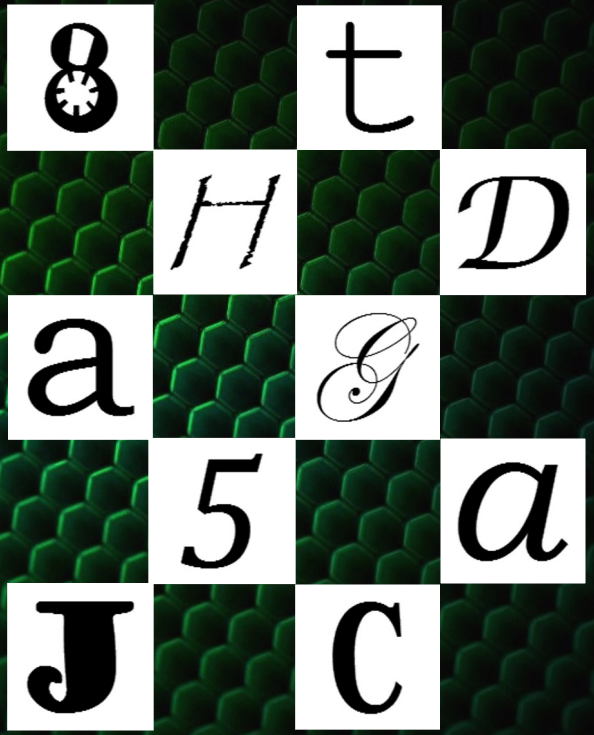
Born-Digital Images

- Generated from Web and Emails such as headings, advertisement etc.
- for Robust Reading competition
- Low-resolution with digitally created Text
- Contains total 541 images
- Mostly regular texts

Testing Datasets

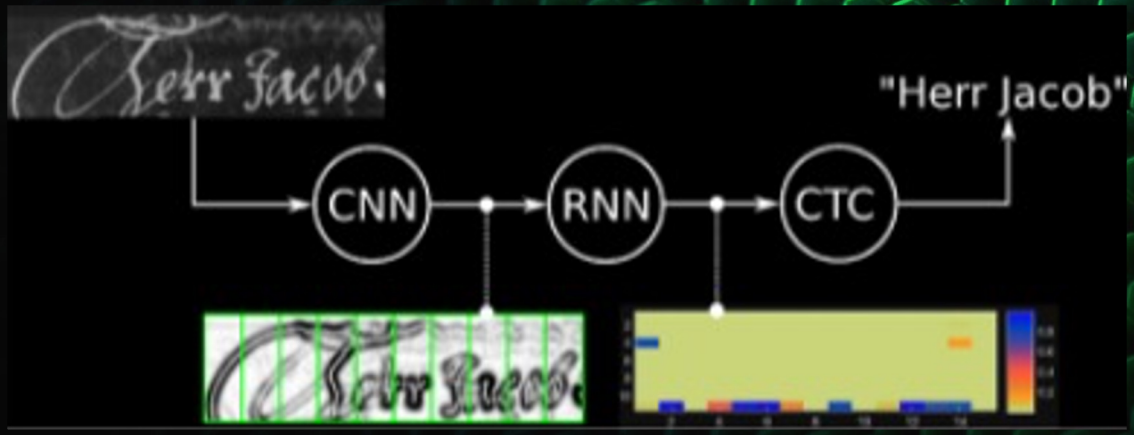
English Character Dataset

- Generated by using of different font styles
- Total 62 batches in 3 different classes
  - Capital character - A to Z (26)
  - Small character - a to z (26)
  - Numbers - 0 to 9 (10)
- Total 62992 images with 1016 images for each batch
- Mostly regular character images



Connectionist Temporal Classification (CTC)

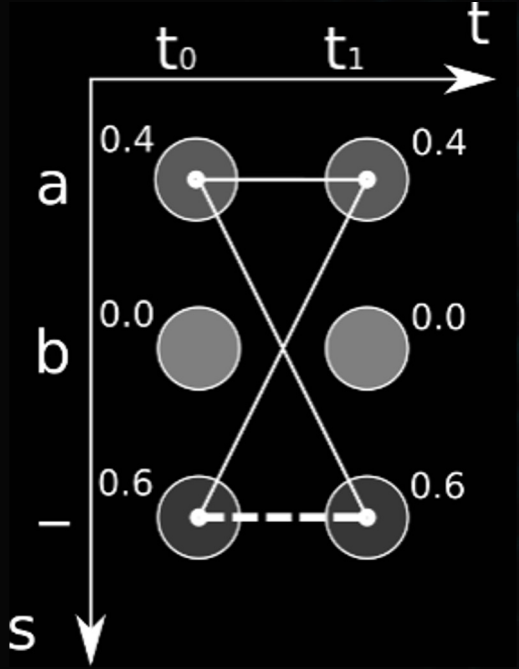
- Well-suited for text and speech recognition
- Alignment-free loss function allowing character repetition in output
- CTC Loss takes the sum of probabilities of all possible alignments



How CTC Decoding Works?

CTC Loss function

Takes the negative of sum of log of probability score for all alignments



$$\textbf{Probability Score of ``a ''} = 0.4 \times 0.4 + 0.4 \times 0.6 + 0.6 \times 0.4 = 0.64$$

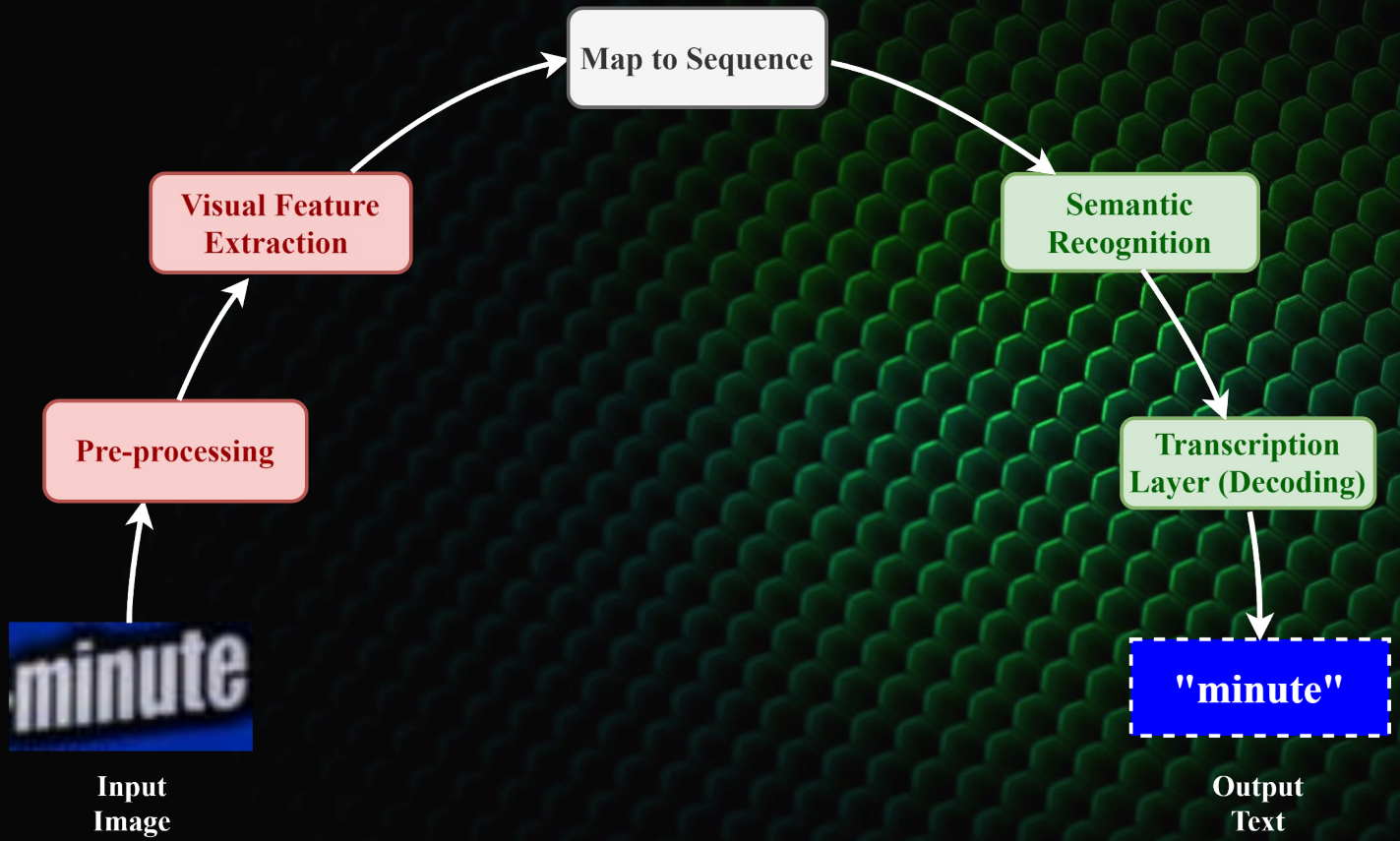
$$\textbf{Probability Score of ``b ''} = 0.0 \times 0.4 + 0.6 \times 0.0 + 0.0 \times 0.0 = 0.0$$

$$\textbf{Probability score of ``– ''} = 0.6 \times 0.6 = 0.36$$

If the Ground Truth is “a”, less penalization.

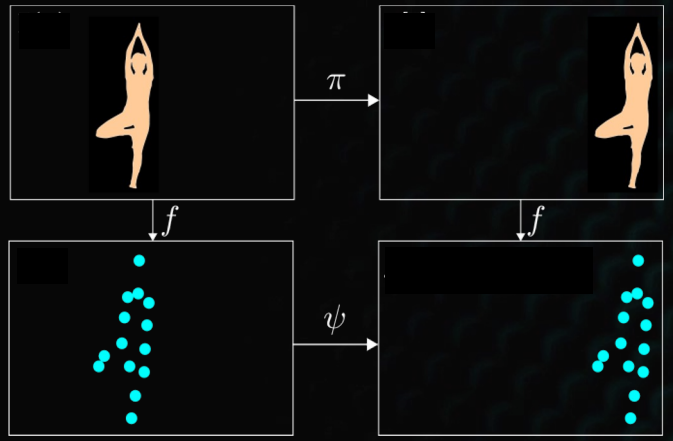
If it is anything else, more penalization

Top Level Block diagram



Underlying Elemental Blocks

Visual Feature Extraction Stage



➔ Typically, Deep Convolutional Neural Networks are used to extract features *automatically*, due to the following advantages :

- Sparse Connectivity → Less Tendency to Overfit
- Translational Invariance → Due to the use of MaxPool Layers
- Weight Sharing → Capability to extract intrinsic “features” or patterns with good generalization

Visual Feature Extraction Stage : (1) Basic CNN

➔ 7 layered Fully Convolutional Neural Network

➔ Basic Operations involved are :

- 2D Convolutions
- 2D Max Pooling
- 2D Batch Normalization

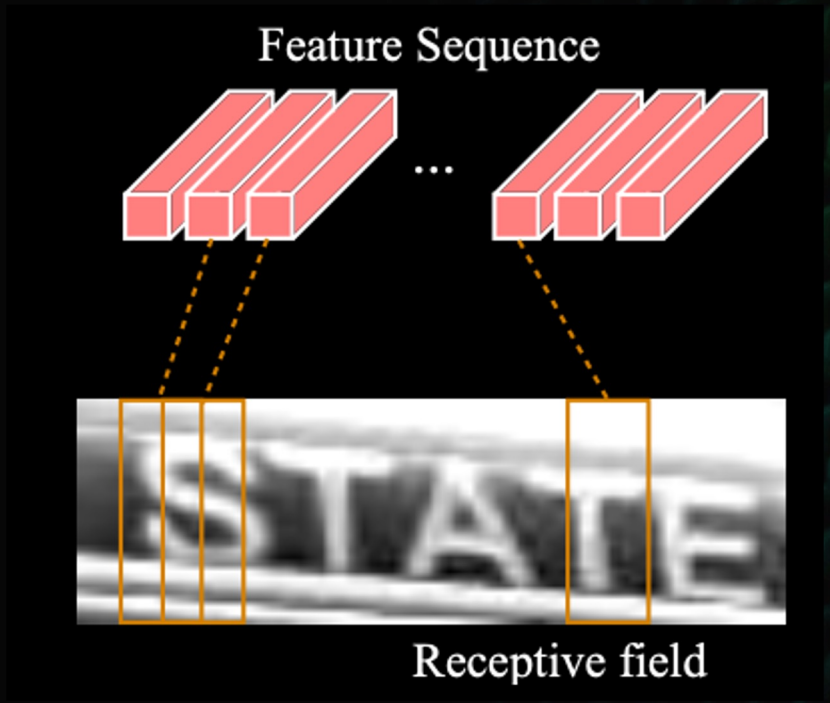
➔ ReLU Activation *(after each Conv2D layer)*

Batch Normalization is used to ensure smoother training.

Convolution	#maps:512, k:2 × 2, s:1, p:0
MaxPooling	Window:1 × 2, s:2
BatchNormalization	-
Convolution	#maps:512, k:3 × 3, s:1, p:1
BatchNormalization	-
Convolution	#maps:512, k:3 × 3, s:1, p:1
MaxPooling	Window:1 × 2, s:2
Convolution	#maps:256, k:3 × 3, s:1, p:1
Convolution	#maps:256, k:3 × 3, s:1, p:1
MaxPooling	Window:2 × 2, s:2
Convolution	#maps:128, k:3 × 3, s:1, p:1
MaxPooling	Window:2 × 2, s:2
Convolution	#maps:64, k:3 × 3, s:1, p:1
Input	W × 32 gray-scale image

https://distill.pub/2017/ctc/

Visual Feature Extraction Stage : (1) Basic CNN



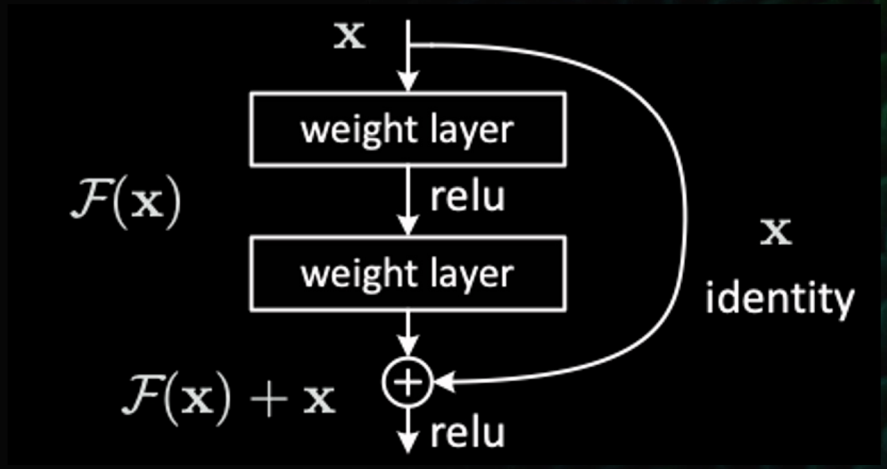
- ➔ Sequence of Feature vectors generated from left to right *(Map to Sequence)*
- ➔ Each vector corresponds is a descriptor of a rectangular ***receptive field*** portion of the image
- ➔ This sequence is the input to the semantic recognition (recurrent) stage

Visual Feature Extraction Stage : (2) ResNet18

- ➔ Modified ResNet18 architecture (popular ImageNet Challenge)
  - 18 Layered (17 + 1 Bridge) Fully Convolutional NN
- ➔ Basic Operations involved are :
  - 2D Convolutions
  - 2D Max Pooling
  - 2D Batch Normalization
  - Residual “Skip” Connections
- ➔ BatchNorm2D+ReLU Activation *(after each Conv2D layer)*
- ➔ Output interpreted as sequence of Feature vectors generated from left to right, fed into next stage.

Layer Name	Output Size	Modified ResNet18
conv1	$112 \times 112 \times 64$	$7 \times 7, 64$, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2 $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
Bridge conv6	$16 \times 1 \times 512$	$[4 \times 4, 512] \times 1$

Visual Feature Extraction Stage : (2) ResNet18



➔ Advantages of Residual Learning :

- Solves the Vanishing Gradients Problem
  (due to skip connections, gradients flow back deep!)
- Avoid Deep “Hidden” Representations excessively (& non-linearly) morphed

So, instead of say $\mathrm{H}(\mathrm{x})$, initial mapping, let the network fit,

$F(x) := H(x) - x$ which gives $H(x) := F(x) + x$.

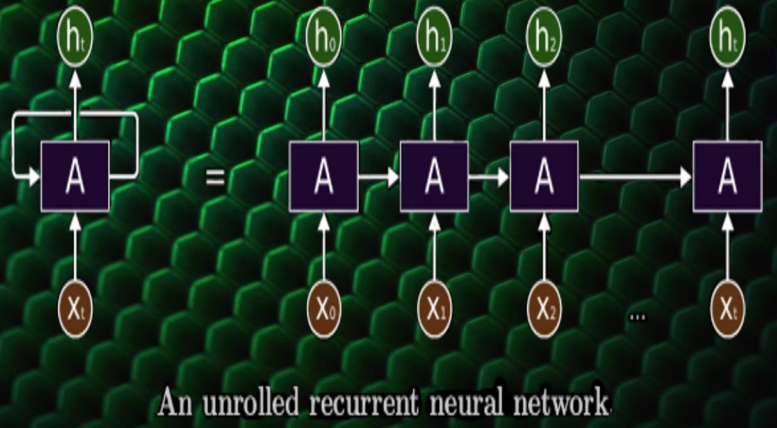
Semantic Recognition Stage

➔ Typically, Recurrent Neural Networks (RNNs) are used to predict a conditional label distribution $p(y_t|c_t)$, for each input feature vector c_t:

- Capture Contextual Information in a sequence
- Capable of operating on sequences of arbitrary lengths, traversing from <SOS> to <EOS>.
- Backpropagates error differentials to input

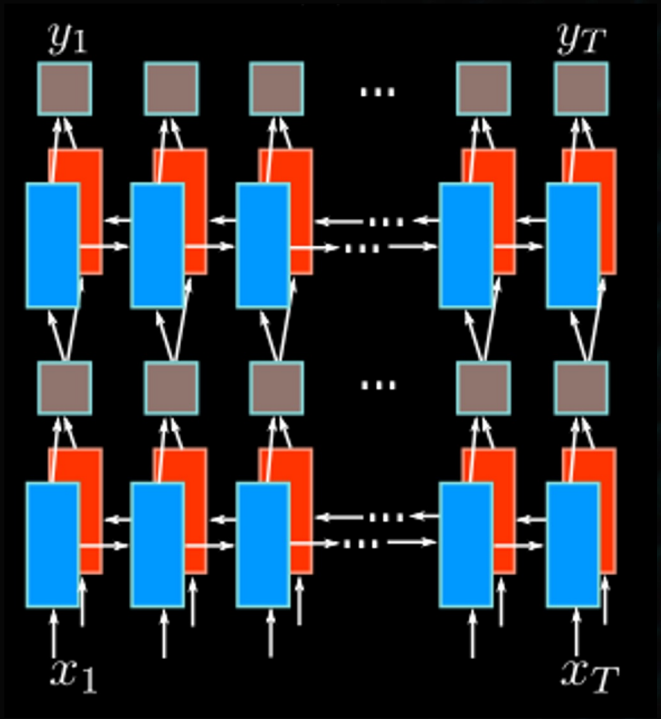
↓

Joint Training of CNN+RNN



An unrolled recurrent neural network

Semantic Recognition Stage : BiLSTM



➔ Two Layered (Stacked) Bidirectional LSTM
(Long Short-Term Memory Networks)

- No. of Hidden Units = 256
(per layer per direction)

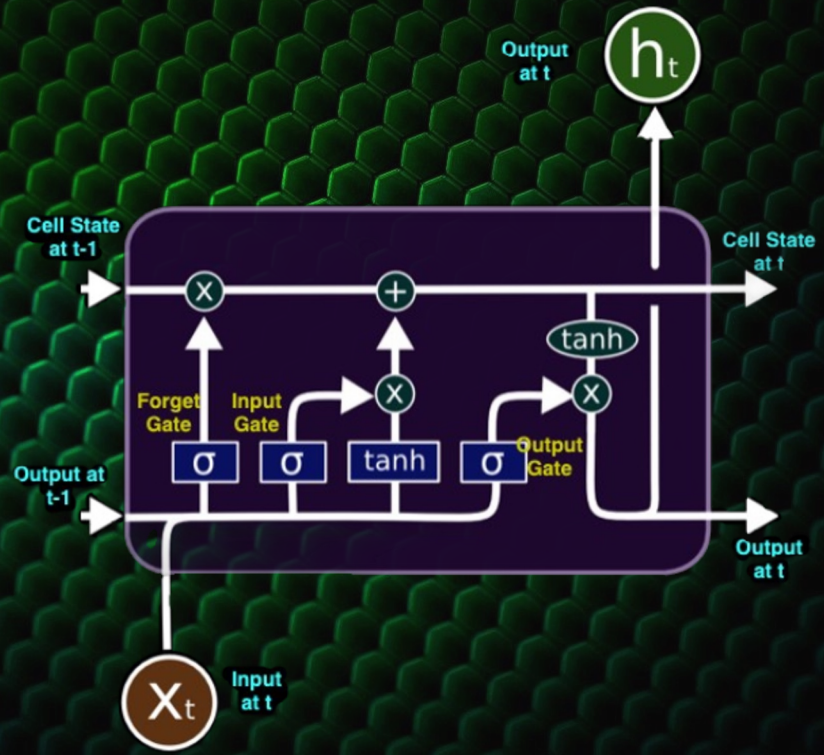
➔ Error Differentials are backpropagated using BPTT

- “Sequences” of Error Differentials concatenated to maps for using BackProp. to train lower layers
(Map to Sequence operation is inverted)

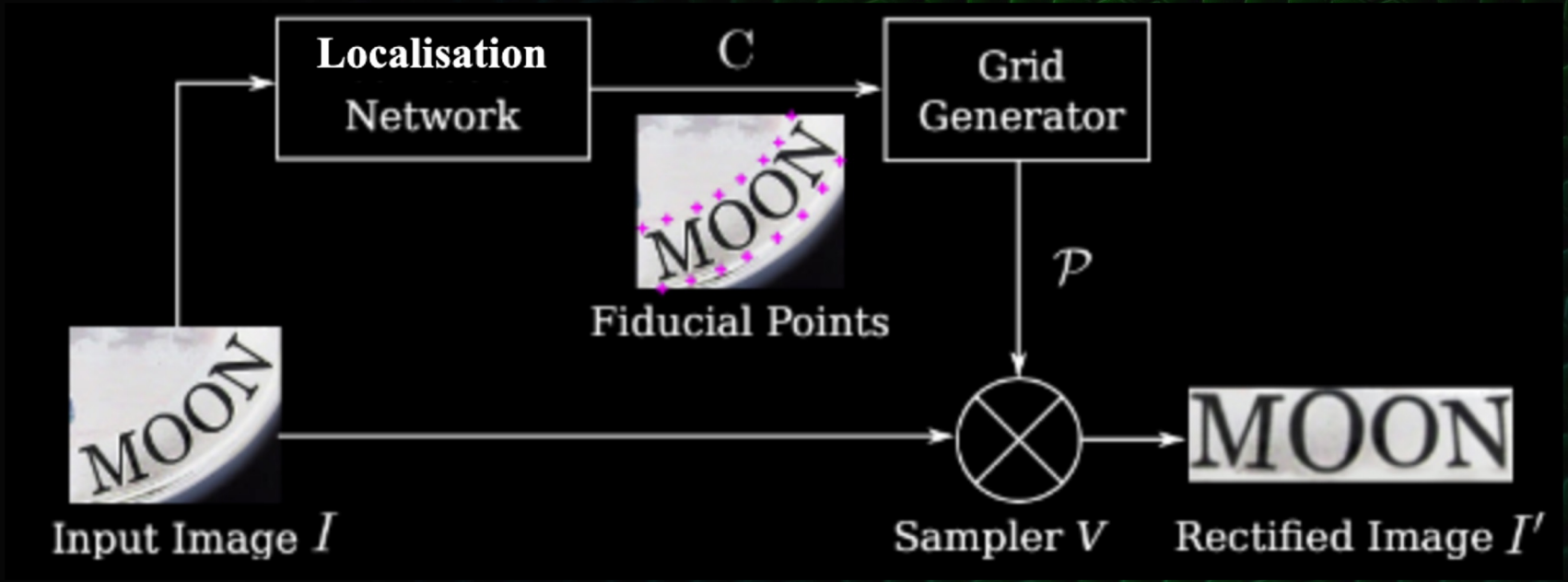
Semantic Recognition Stage : BiLSTM

➔ Advantages compared to other RNN's:

- Long-Term Dependencies are *efficiently* captured
- Selective {Read,Write, Forget} operations using multiple gates (input, output, forget gates resp.)
- Solves the Vanishing Gradients Problem !
- Bidirectional → Useful Context from both sides!



Pre-Correction Stage : Spatial Transformation Network (STN)

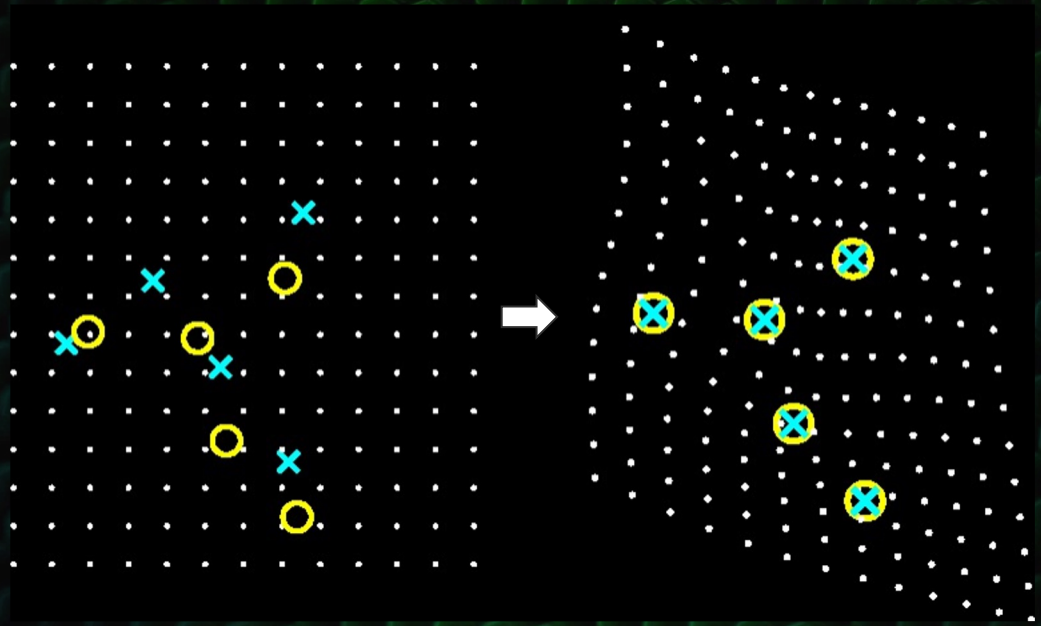


Pre-Correction Stage : Spatial Transformation Network (STN)

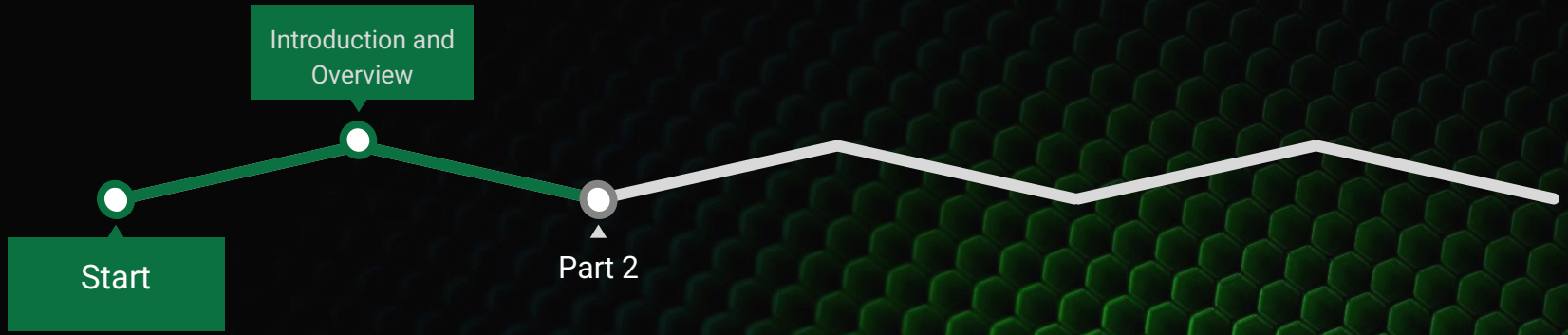
- ➔ Localisation Network - Locates K(even) fiducial points in the image
- ➔ Grid Generator - Estimates the TPS transformation parameters
- ➔ Sampler - Interpolate the pixel values in the output from the corresponding neighbourhood of pixels in the input.

TPS(Thin Plate Spline) Transformation

- Smooth interpolation of points with infinite order differentiability
- Resistant to bending (Like a thin sheet of metal)
- Smoothness can be increased by regularization



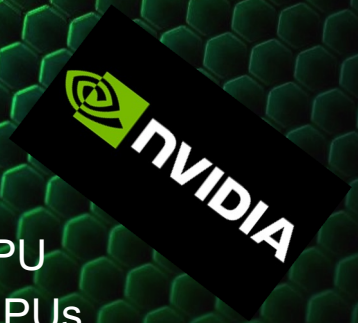
TPS(Thin Plate Spline) Transformation



Implementation Details

General Details

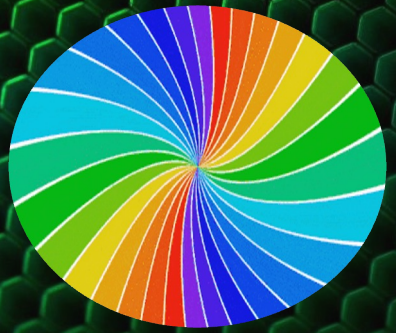
- Framework Used : PyTorch (For Training,Validation & Testing)



- Device Configuration :
  - For Training & Validation → NVIDIA RTX GeForce 2060 GPU
  - For Testing on Various Datasets → Google Colab's Tesla GPUs

Preprocessing Details

- PyTorch's Lightning Data Collator and Data Transformer was used for the following preprocessing steps, wherever required *(Specifics are mentioned in later slides)* :
  - Data Loading for Training,Validation and Testing
  - RGB ⟷ Grayscale Conversion
  - Resizing, Converting to Tensors & Normalizing Images

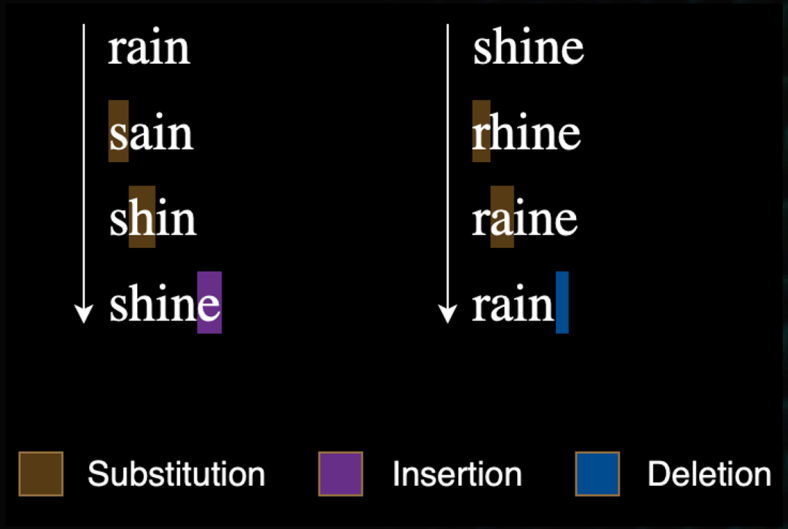


- Batching → For Larger Models, 32 images/batch whereas for smaller models 64 images/batch

Training Configuration

- Output Classes *(of RNN)* → 81 Classes (26+26+10+19)
- **Loss Function** (CTC) Loss → Connectionist Temporal Classification
- **Optimizer** → Adam Optimizer (lr = 0.001)
- **Checkpointing** of Models → Decrease in validation loss
- **Train - Val Split** → 80 % - 20 %

Model Evaluation Metrics : Character & Word Accuracy

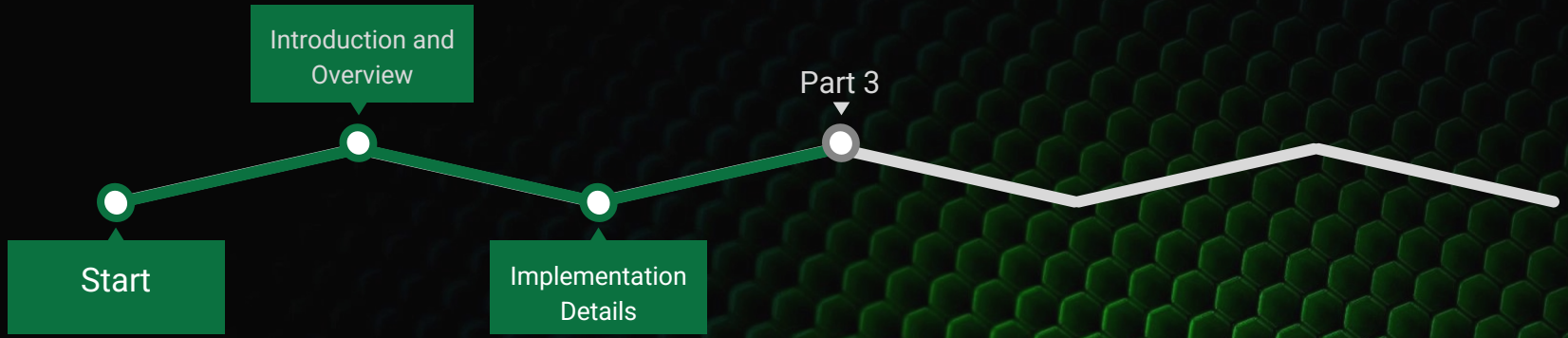


Lev("rain","shine") = 3

Levenshtein Distance is the min. of operations to be performed to convert one string to another

$$CA(pred, truth) = \left(1 - \frac{Lev(pred, truth)}{|truth|}\right) \times 100\%$$

$$WA(pred, truth) = (pred == truth) \times 100\%$$



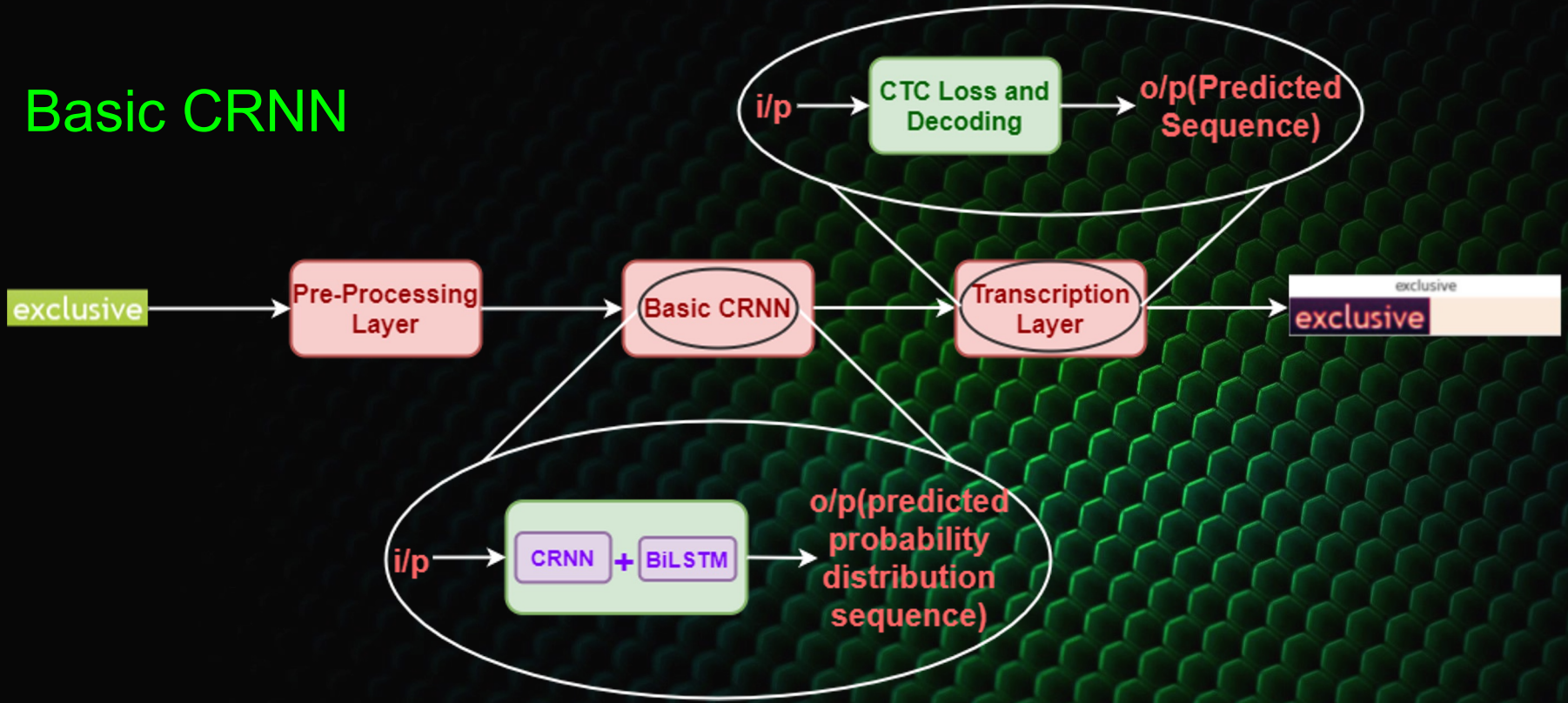
Model Design

Stage 1.0 : Regular Text Recognition

Training Details :

- Training Data Size : 527632
- Val Data Size : 131909
- Batch Size : 64
- Iterations (or) Batches (or) Updates per epoch : 8245
- No. of Epochs : 6
- Average rate : 1.43 it/s
- Total Time (For Training) : ~12.5 hrs

Basic CRNN

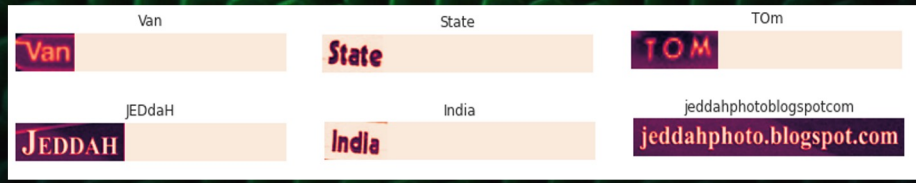


Number of Trainable Parameters: 8.35×10^6

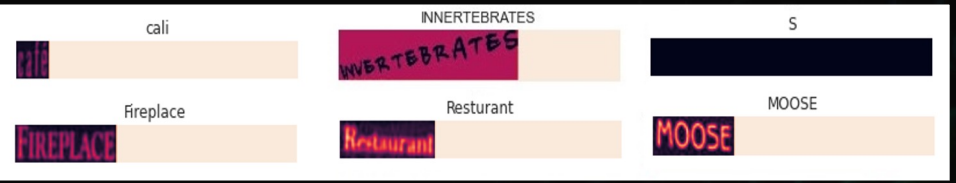
Results

	Synth90k	IIIT5k	SVTP	ICDAR13	ICDAR15	Born-Digital Images	CUTE 80
CA	91.86	81.44	47.13	85.05	60.45	85.29	45.28
WA	78.74	65.55	26.51	73.24	36.18	72.1	39.37

Result for Synth90k dataset

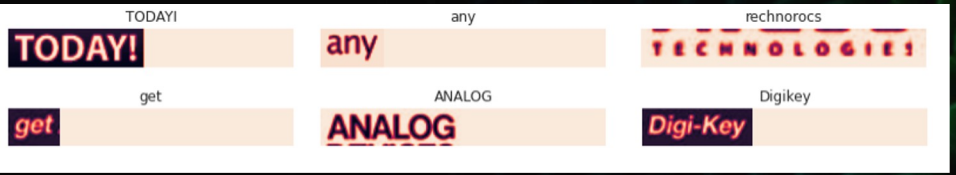


Result for IIIT5k dataset



Result for STVP dataset

Result for ICDAR13 dataset

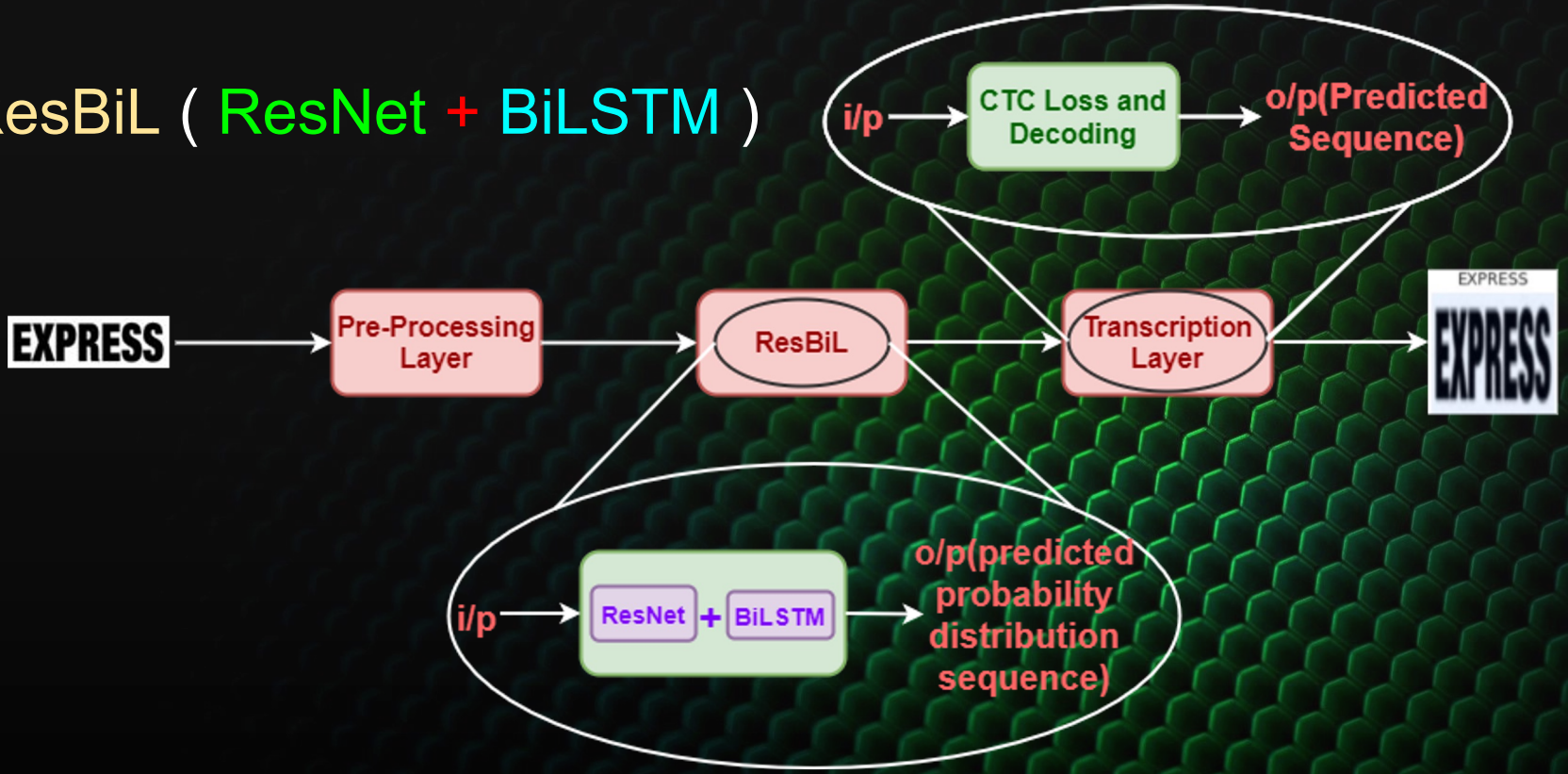


Result for ICDAR15 dataset

Result for Born Digital Images dataset

Result for CUTE 80 dataset

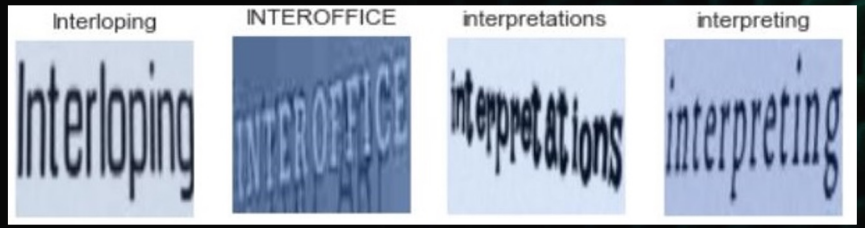
ResBiL (ResNet + BiLSTM)



Number of Trainable Parameters: 1.82×10^7

Results

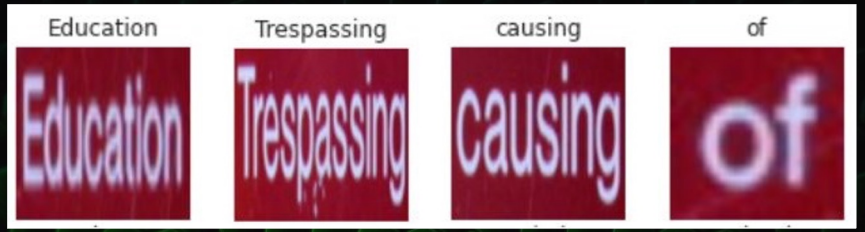
	Synth90k	IIIT5k	SVTP	ICDAR13	ICDAR15	Born-Digital Images	CUTE 80
CA	91.74	84.29	52.31	84	64.81	86.7	46.73
WA	77.11	67.77	30.69	70.51	39.92	72.14	40.28



Result for Synth90k dataset

Result for IIIT5k dataset

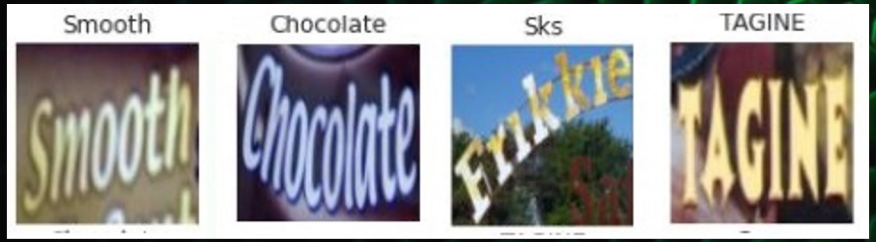
Result for STVP dataset



Result for ICDAR13 dataset

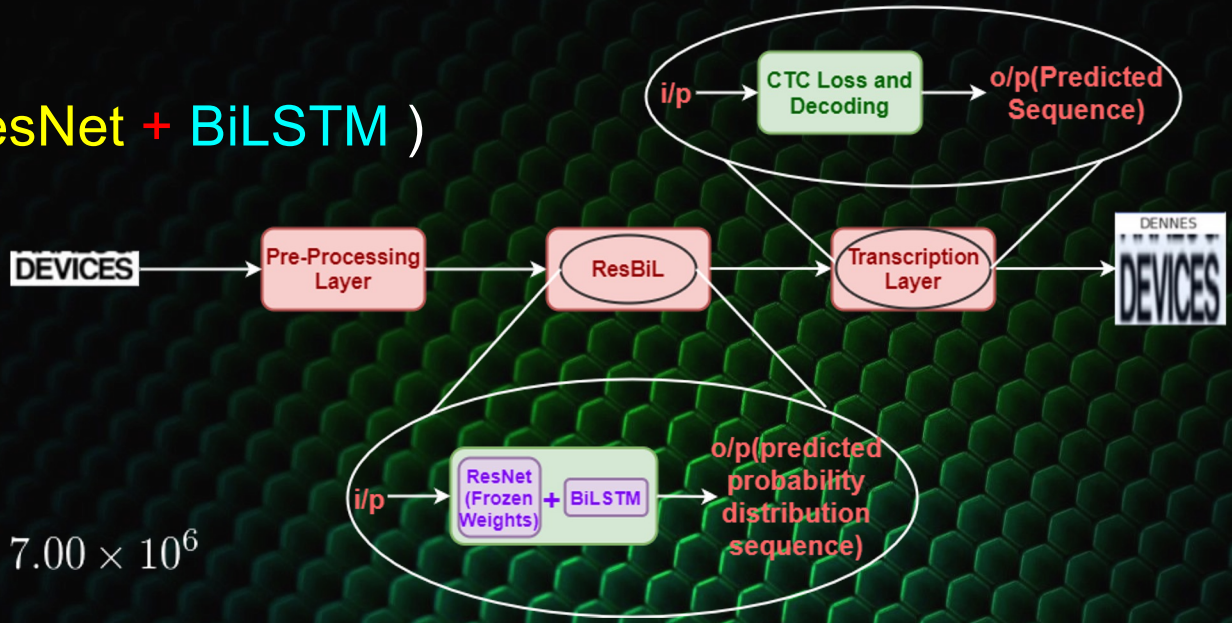
Result for ICDAR15 dataset

Result for Born Digital Images dataset



Result for CUTE 80 dataset

FResBiL (Frozen ResNet + BiLSTM)



Number of Trainable Parameters: 7.00×10^6

Results

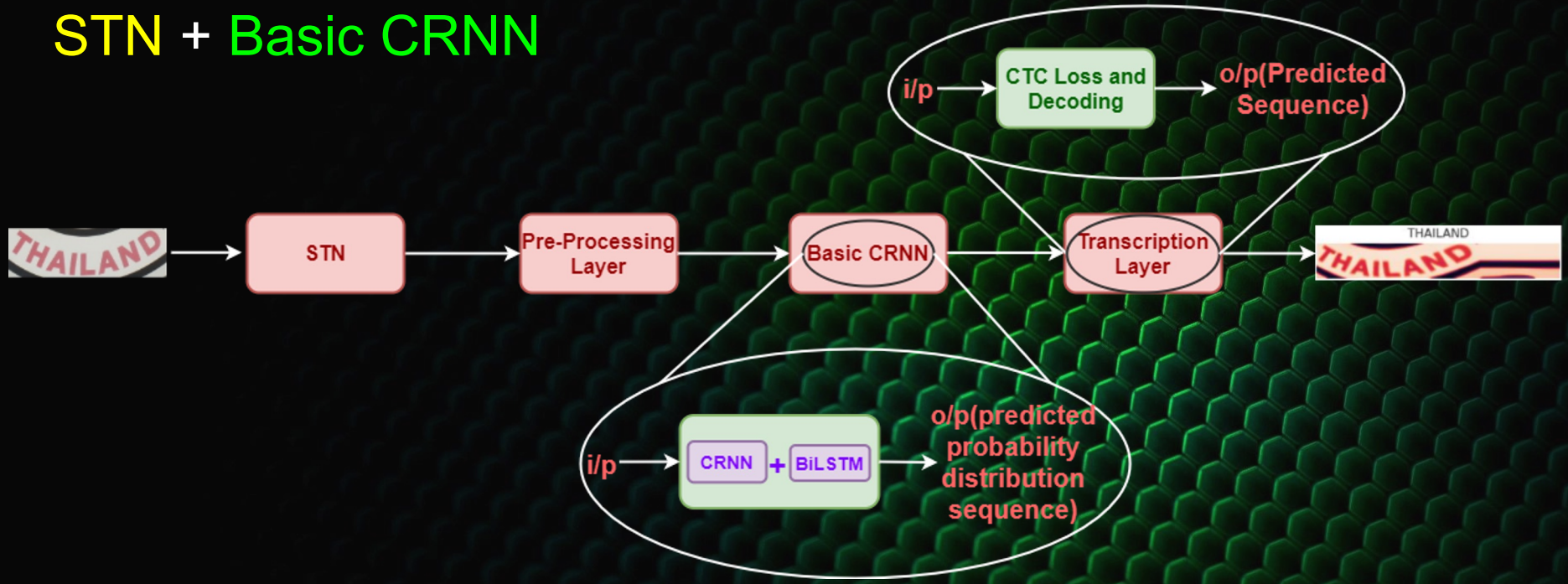
	Synth90k	IIIT5k	SVTP	ICDAR13	ICDAR15	Born-Digital Images	CUTE 80
CA	44.33	13.56	6.64	9.6	8.23	17.46	7.06
WA	3.26	1.52	0	0	0	1.69	0.35

Stage 2.0 : Irregular Text Recognition

Training configuration:

- Training Data Size : 527632
- Val Data Size : 131909
- Batch Size : 32
- Iterations (or) Batches (or) Updates per epoch : 16489
- No. of Epochs : 6
- Average rate : 2.70 it/s
- Total Time (For Training) : ~13.5 hrs

STN + Basic CRNN



Number of Trainable Parameters: 1.00×10^7

Results

	Synth90k	IIIT5k	SVTP	ICDAR13	ICDAR15	Born-Digital Images	CUTE 80
CA	91.79	83.78	52.59	85.13	66.49	86.38	63.32
WA	78.41	67.41	32.4	73.42	39.84	71.26	38.19

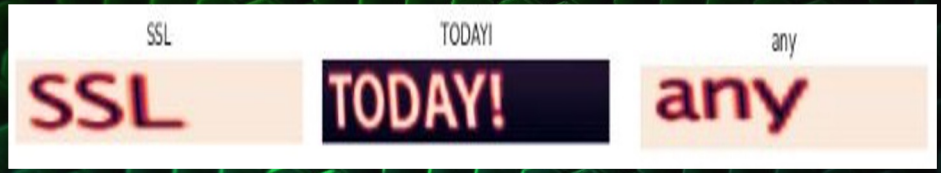
Result for Synth90k dataset

Result for IIIT5k dataset

Result for STVP dataset

Result for ICDAR13 dataset

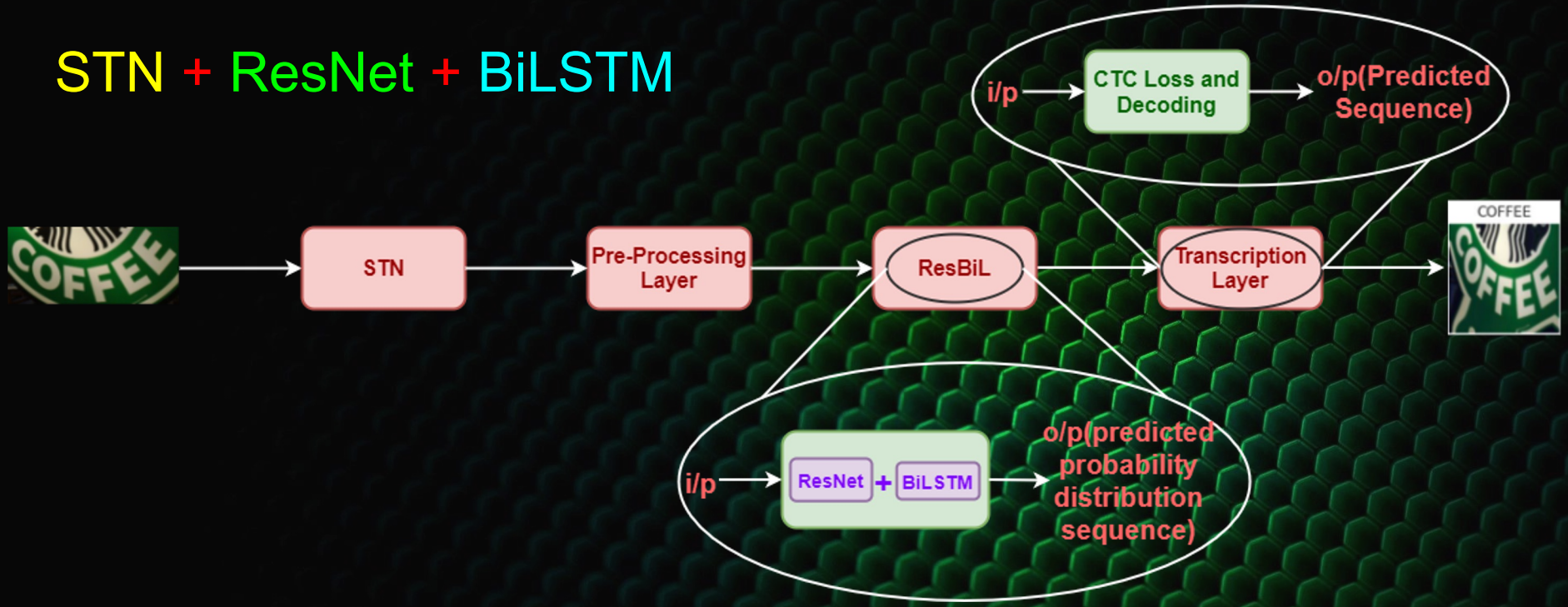
Result for ICDAR15 dataset



Result for Born Digital Images dataset

Result for CUTE 80 dataset

STN + ResNet + BiLSTM

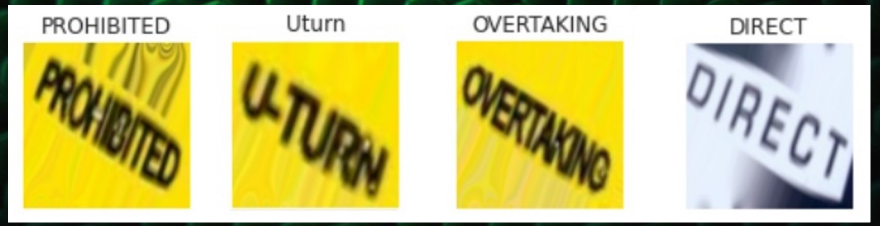


Number of Trainable Parameters: 1.99×10^7

Results

	Synth90k	IIIT5k	SVTP	ICDAR13	ICDAR15	Born-Digital Images	CUTE 80
CA	92.45	83.48	52.59	83.59	65.91	85.9	65.57
WA	79.92	66.81	34.73	71.16	41.17	71.39	40.28

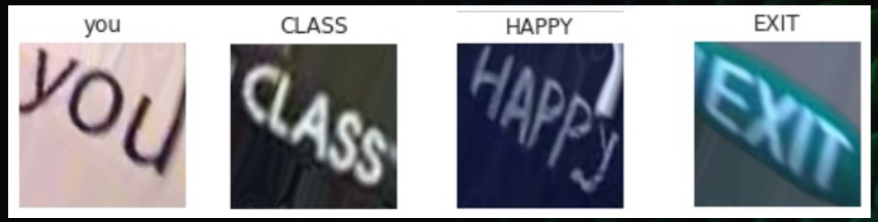
Result for Synth90k dataset



Result for IIIT5k dataset

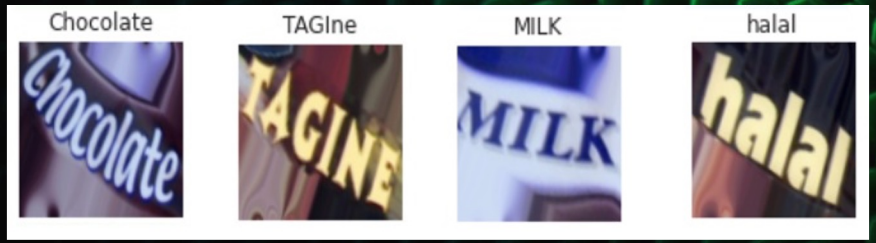
Result for STVP dataset

Result for ICDAR13 dataset

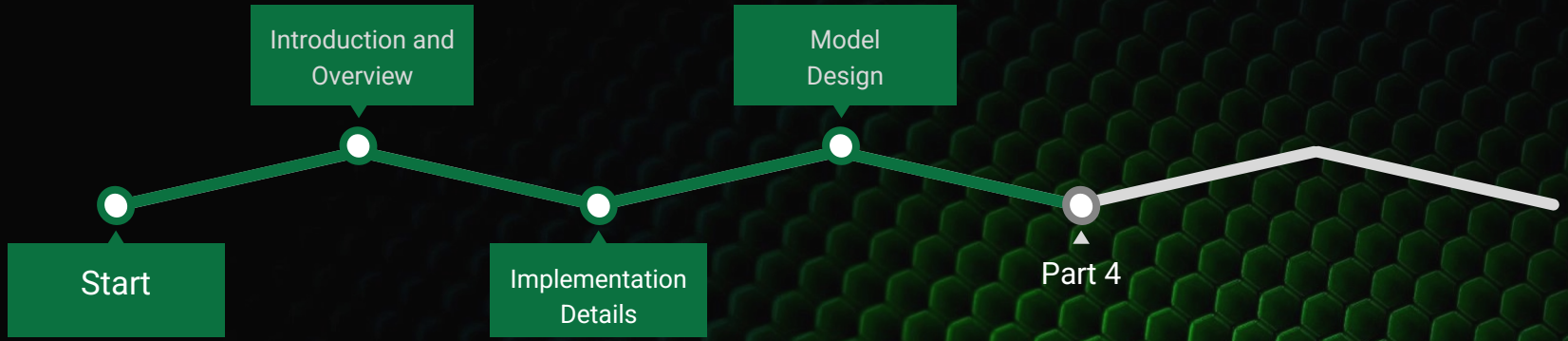


Result for ICDAR15 dataset

Result for Born Digital Images dataset



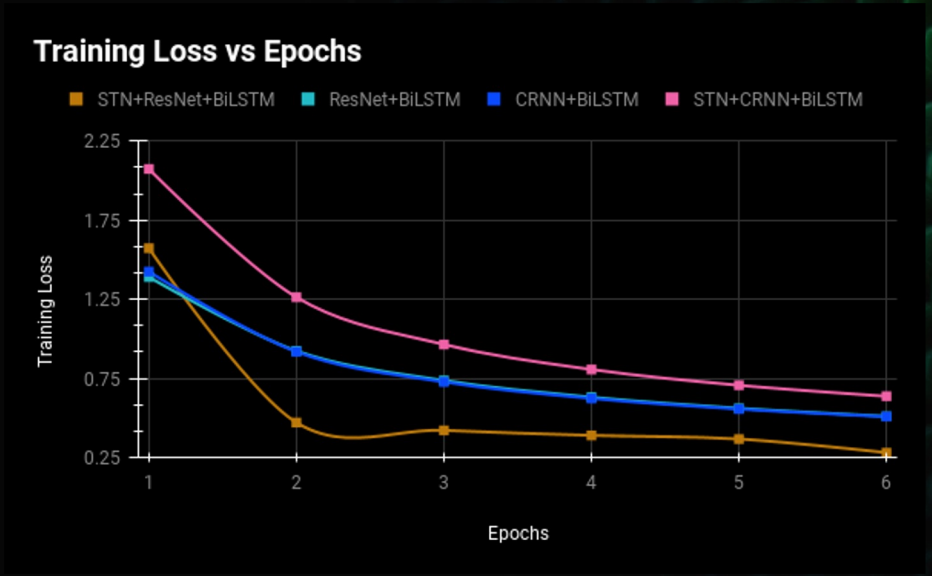
Result for CUTE 80 dataset

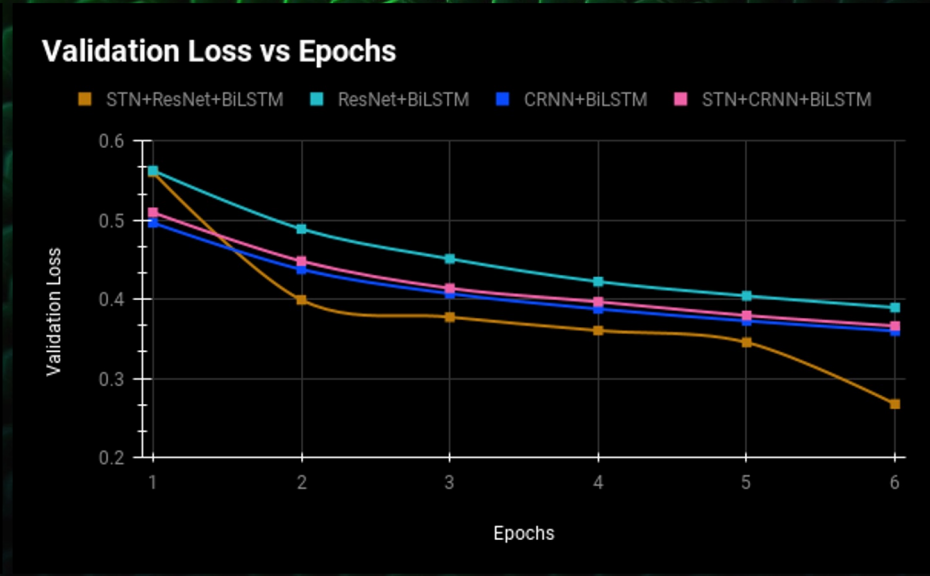

Introduction and Overview
Model Design
Start
Implementation Details
Part 4

Analysis

Training Curves : Training and Validation Loss

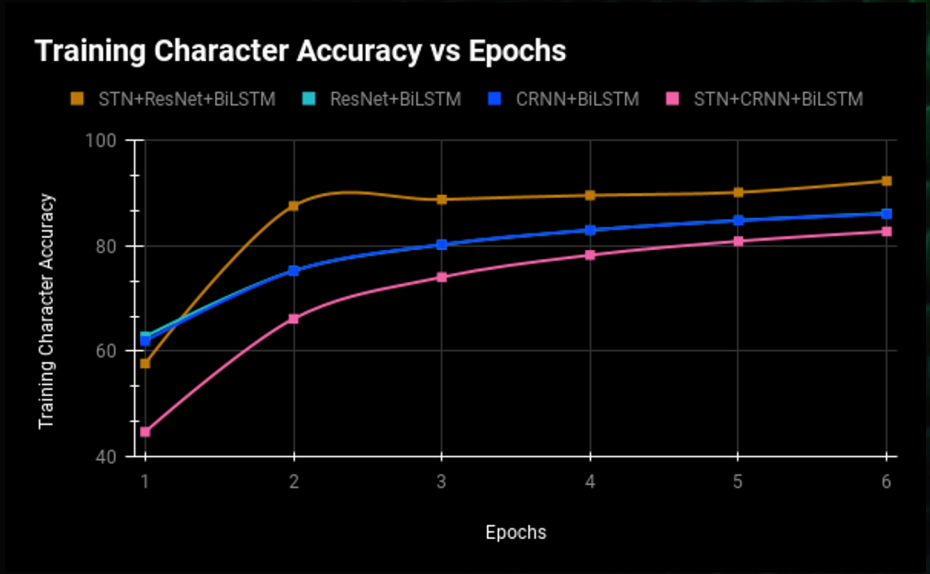
- ➔ Significant drop in training loss between 1st and 2nd epoch for all the models
- ➔ STN+ResNet+BiLSTM has the best performance among all these models
- ➔ High loss at the initial epochs maybe due to different initializations
- ➔ More number of epochs might have yielded a better results as inferred from Validation loss
- ➔ Since both are decreasing we haven't still reached overfitting

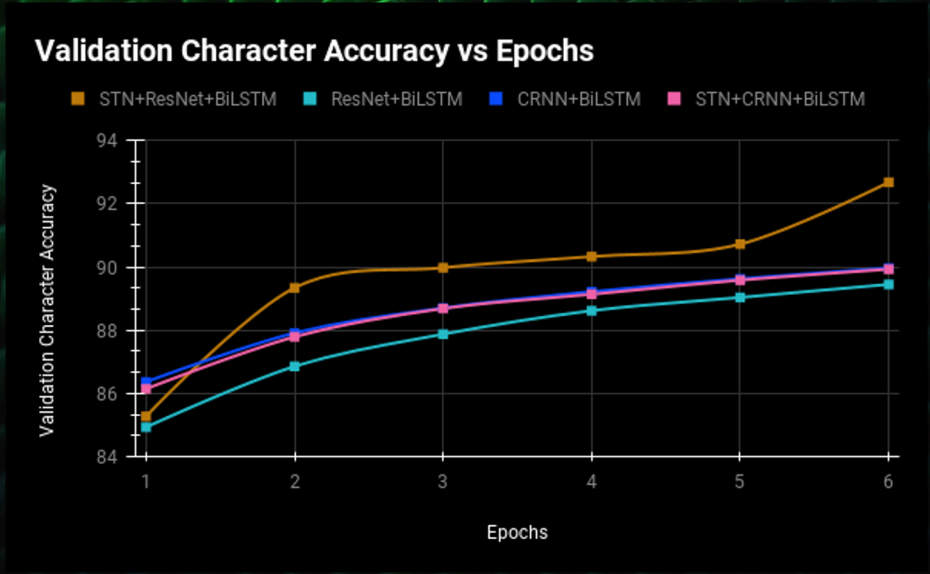




Training Curves : Training and Validation CA

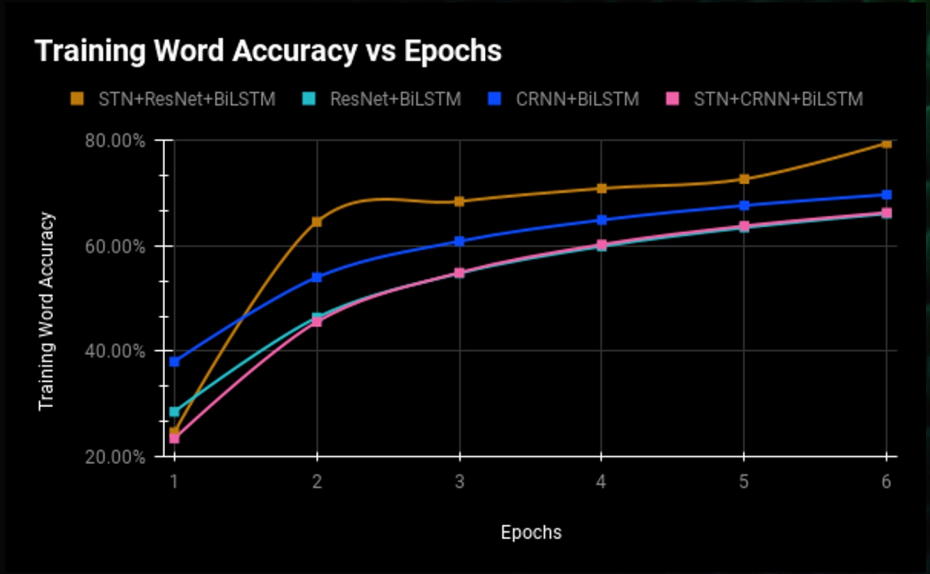
- ➔ Character Accuracy trend supports the Loss trend
- ➔ Again, more number of training data and more epochs might have been better
- ➔ High learning for characters at the initial epochs
- ➔ Training and Validation CA doesn't show much deviation after 2nd epoch

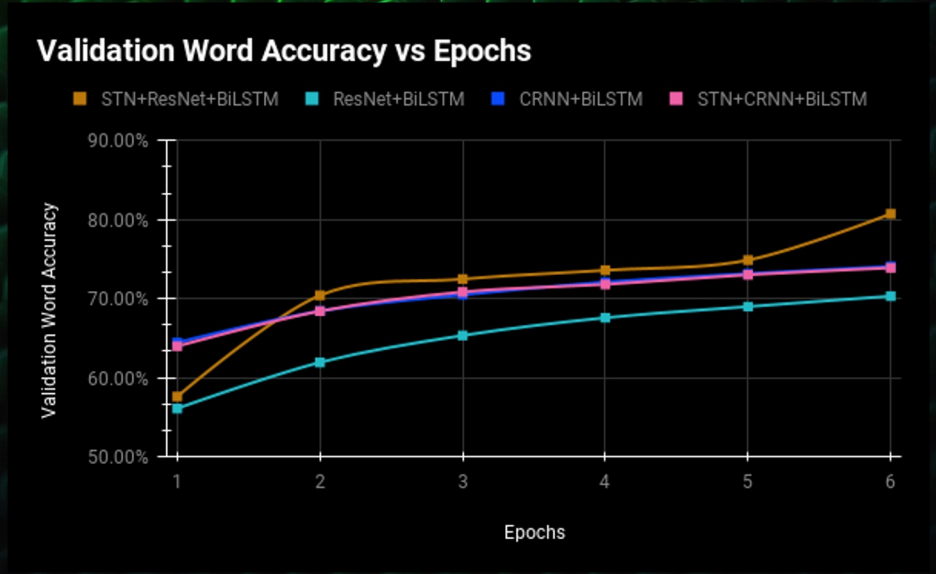




Training Curves : Training and Validation WA

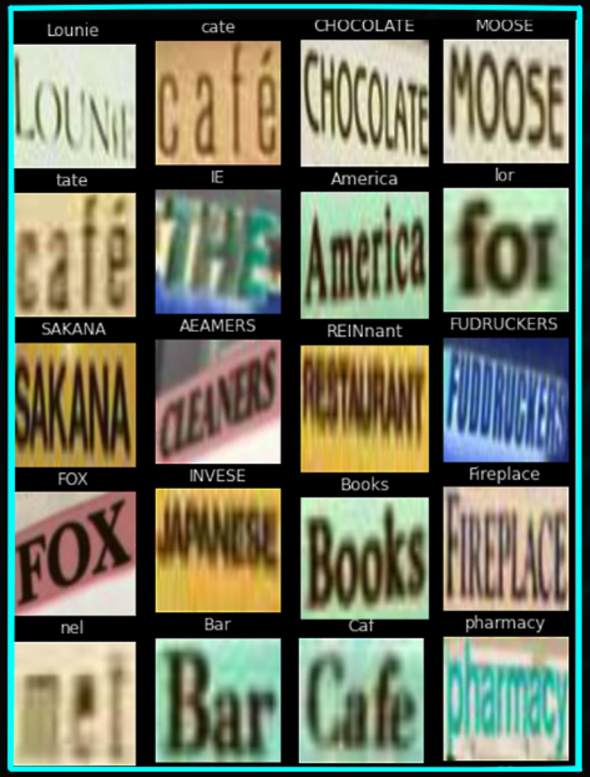
- ➔ Word Accuracy also follows same trend as Character Accuracy
- ➔ Word accuracy has increased between 1st and 2nd epoch but still is low
- ➔ Dramatic increase of WA at the end epochs for STN+ResNet+BiLSTM model
- ➔ Reason maybe the model is learning character at it's best first and later learning the sequencing





Interesting Observations : Pre-Correction

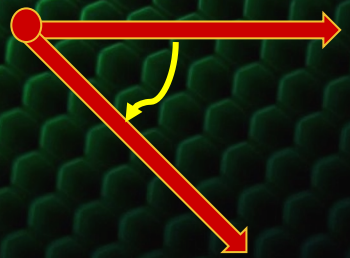
ResNet + BiLSTM



STN + ResNet + BiLSTM

Orientation Correction is handled well

Mean Orientation “learnt” from End2End Training



Interesting Observations : Pre-Correction

Selective Emphasis

↓

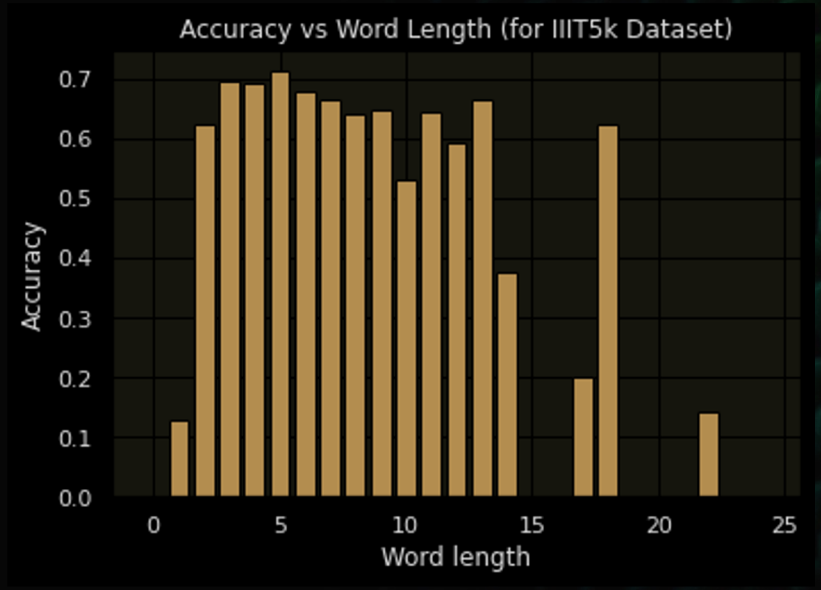
Partially visible semantics into noise

+

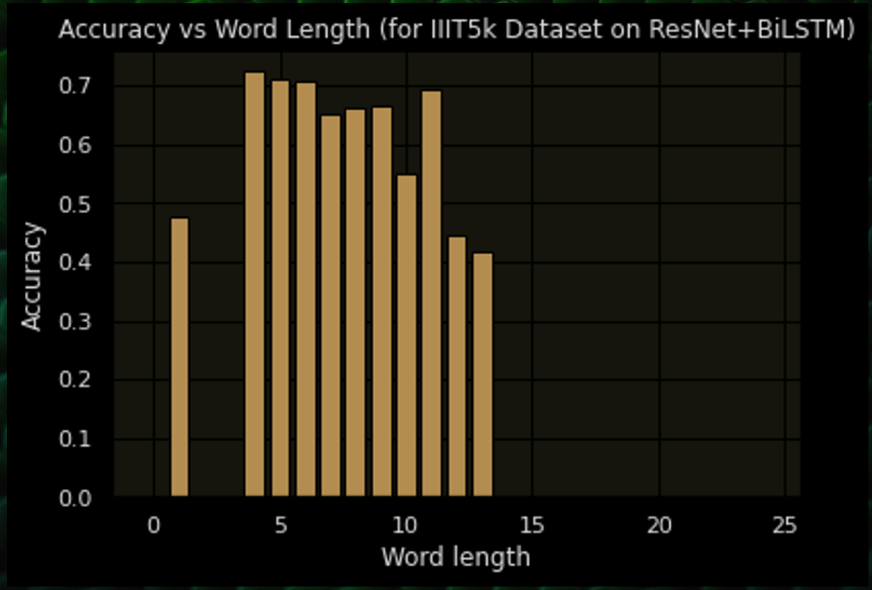
Clearly visible semantics are stressed

Accuracy vs Word Length

- Analysis of IIIT5k dataset on both models without STN



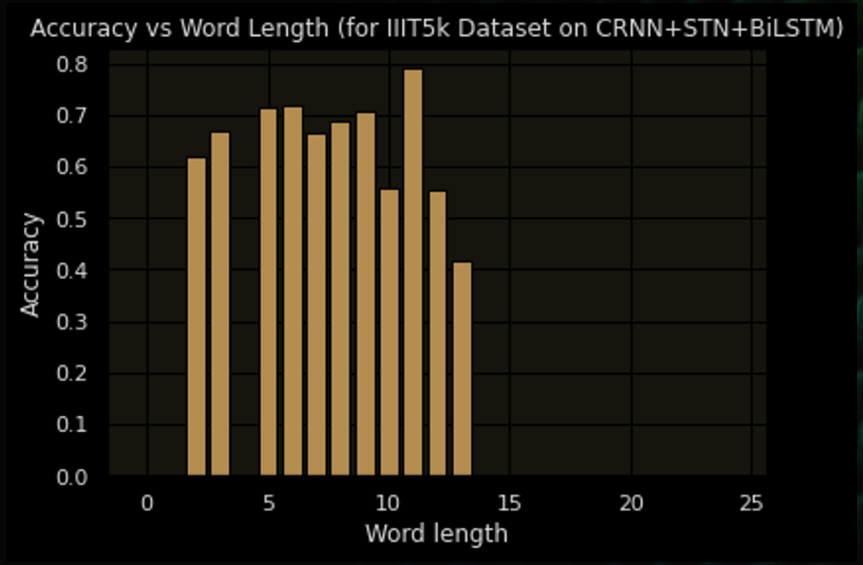
Tested on Basic CRNN



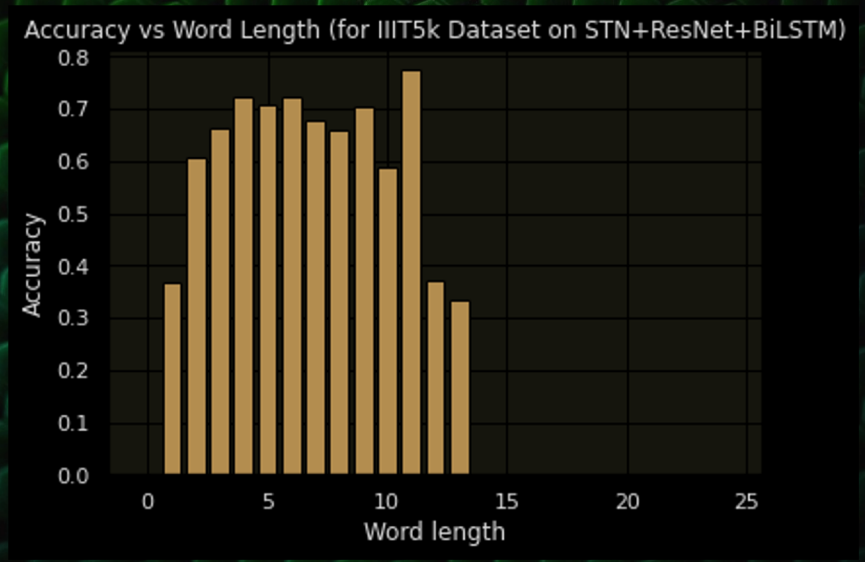
Tested on ResNet+BiLSTM

Accuracy vs Word Length

- Analysis of IIIT5k dataset on both models with STN

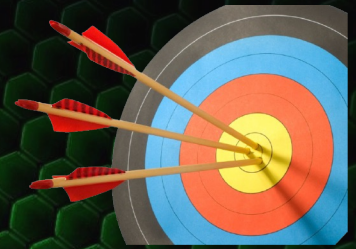


Tested on STN+ CRNN



Tested on STN+ResNet+BiLSTM

Accuracy vs Word Length

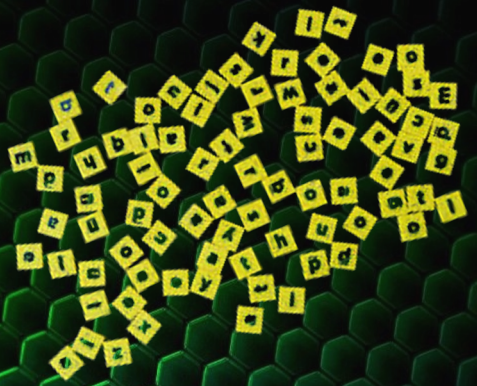


Inferences

- High accuracy for average word length for almost all the model
- CRNN’s good performance over wider range could be because of variable length images
- We think that square size pre-processing image in ResNet model could be a reason for poor performance on wider range
- Improvement in accuracy with STN (greater than 0.7)
- CRNN+STN is length selective:
    - Giving very good results for some lengths and very poor for some other lengths
- STN+ResNet :
    - Again, square processing of images could be restricting performance

Character Confusion Analysis

- As expected diagonal is perfectly aligned



- Prone to human errors

- **Poor predictions :**
  - ℓ →
  - z, 7, ℓ →
  - 0(Zero) →
  - 7 →

T (t)
o
z

- **Comparable Predictions :**
  - I R (r) → p
  - m → n
  - q → 9
  - D → O
  - Z → 7

- Most of the letters having only one horizontal bar at top(like I,J,7 not E.F) are predicted as I.
- Most of the letters having only one Vertical bar at top(like 1 and l(Lowercase L)) are predicted as l.

True Labels

Predicted Labels	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	1	2	3	4	5	6	7	8	9	Sum
a	570	0	0	2	0	0	0	0	0	0	0	0	6	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	580
b	0	1659	0	215	2	2	1	10	0	0	2	0	1	2	1	1	0	6	0	0	0	3	0	0	0	0	0	0	0	5	0	2	2	0	6	0	1920
c	164	5	1220	209	80	5	96	0	22	1	3	51	0	0	42	1	53	5	29	71	94	19	12	3	2	167	20	2	275	1	0	0	4	0	0	1	2657
d	0	6	0	908	0	0	1	0	0	0	0	0	3	12	33	23	0	0	1	0	4	9	1	0	0	0	19	0	2	0	0	0	0	0	0	0	1022
e	31	57	17	0	1157	543	8	121	10	0	40	19	1	2	7	16	8	39	13	19	1	2	7	10	0	3	1	0	2	4	3	21	28	2	9	2	2203
f	10	23	5	0	18	1313	22	366	35	11	16	7	2	21	2	55	13	160	12	54	1	8	2	6	14	11	2	1	0	3	18	54	4	5	8	4	2286
g	54	9	713	34	574	1	1057	4	12	6	3	5	0	1	152	2	299	8	59	5	3	4	1	2	9	18	24	0	3	1	7	2	35	0	30	24	3161
h	0	6	0	0	0	0	0	702	0	0	2	0	14	4	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	734
i	70	51	18	110	20	38	33	331	964	82	114	1116	78	307	16	75	23	193	21	112	71	34	100	29	24	119	8	267	21	6	84	8	1	21	1	2	4568
j	4	0	0	6	0	10	6	4	336	1327	0	6	3	4	1	11	3	2	2	0	1	2	11	1	13	3	0	1	0	4	17	7	0	0	0	4	1789
k	0	1	0	0	0	0	0	55	0	0	1668	0	4	3	0	1	0	16	0	0	0	0	2	5	0	0	0	0	0	0	0	0	0	0	0	0	1755
l	1	16	0	6	0	0	1	154	0	0	0	437	2	1	4	0	4	0	0	0	142	0	10	0	0	0	7	0	0	0	5	0	1	0	0	0	791
m	0	0	0	0	0	0	0	0	0	0	0	0	725	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	729
n	0	0	0	0	0	0	0	0	0	0	0	0	616	804	0	0	0	0	0	0	0	0	42	0	2	0	0	0	0	0	0	0	0	0	0	0	1464
o	4	0	0	62	2	2	1	0	0	2	0	0	0	0	1179	8	4	1	0	0	2	1	1	0	0	0	567	0	1	0	0	0	0	0	2	0	1839
p	16	31	37	111	31	24	18	8	66	32	3	74	36	214	47	1776	64	526	18	17	13	36	8	8	20	16	26	6	102	15	3	0	1	13	24	24	3464
q	36	0	0	41	4	2	1	1	0	1	0	1	0	0	35	0	801	1	0	0	3	6	1	0	5	0	12	0	149	0	1	0	0	0	1	55	1157
r	0	0	0	1	0	0	0	0	1	1	10	2	17	222	0	0	0	741	0	19	0	0	5	0	0	70	0	0	2	0	0	0	0	31	0	0	1122
s	0	0	0	3	1	2	1	1	4	1	2	2	0	0	0	0	1	1	1372	3	0	0	0	3	9	1	0	0	1	16	0	10	0	0	11	0	1445
t	10	1	3	17	15	81	10	44	443	375	19	267	2	41	3	25	30	269	17	1696	36	79	32	101	57	863	0	539	5	2	2	0	0	886	0	1	5971
u	0	0	0	0	0	0	0	0	2	29	0	2	1	0	0	0	0	0	0	2	1354	22	11	0	1	0	0	0	0	0	0	0	0	0	0	0	1424
v	0	0	0	0	0	0	2	2	0	123	0	0	5	20	0	0	0	12	0	0	14	1725	174	0	121	0	0	0	0	0	0	0	0	0	0	0	2198
w	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	4	956	0	0	0	0	0	0	0	0	0	0	0	0	0	964
x	0	0	6	0	0	0	0	7	0	0	5	0	10	2	0	0	0	0	0	0	0	2	17	1715	5	0	0	0	0	0	0	0	0	0	0	0	1769
y	0	0	0	0	0	0	4	4	2	2	3	1	8	15	0	0	1	1	3	3	7	31	73	91	1696	1	0	0	1	2	0	1	0	0	0	0	1950
z	0	0	0	0	1	0	0	0	8	0	19	2	0	0	0	0	0	4	0	3	7	0	0	0	1	203	0	3	0	1	0	0	0	3	0	0	255
0	15	0	0	5	0	0	1	0	0	0	0	0	0	0	417	3	7	0	4	0	5	3	2	1	0	0	288	0	0	0	0	0	0	0	0	0	751
1	6	0	0	33	0	0	1	77	62	4	1	0	20	29	0	1	0	2	1	5	10	5	10	4	0	8	0	115	8	0	1	0	0	1	0	0	404
2	272	0	0	6	4	0	1	1	2	0	0	2	0	0	0	2	1	1	1	0	8	9	8	3	0	6	0	0	215	0	0	0	0	0	0	0	542
3	0	34	2	2	0	0	7	0	4	6	0	1	0	0	0	2	0	0	41	2	1	0	1	2	4	6	0	0	0	842	0	15	0	0	35	0	1007
4	402	3	1	76	0	2	2	9	0	0	2	0	0	1	2	0	18	2	6	0	6	3	1	1	19	4	0	0	4	3	871	1	0	0	0	7	1446
5	0	3	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	1	0	0	0	1	0	884	0	0	0	0	895
6	4	61	1	7	14	0	10	1	0	0	0	0	0	0	10	3	4	0	3	1	2	1	1	0	0	0	9	0	0	0	8	937	0	6	1	1084
7	14	0	0	2	0	0	1	4	31	0	12	19	0	2	0	2	1	12	1	7	5	4	2	24	8	474	0	79	1	3	0	0	0	53	0	2	763
8	29	7	1	1	96	0	90	1	0	0	0	0	0	0	2	0	7	0	298	0	1	0	0	0	0	1	4	0	0	5	0	0	3	0	638	1	1185
9	273	16	7	150	6	2	627	5	24	18	2	12	2	4	73	12	688	9	124	2	3	10	4	4	13	42	29	1	224	101	2	3	0	0	245	888	3625
Sum	1985	1989	2031	2007	2025	2027	2002	1913	2028	2023	1926	2026	1560	1711	2026	2019	2030	2012	2027	2021	1794	2022	1506	2013	2025	2017	1016	1014	1016	1015	1014	1016	1016	1015	1016	1016	60919

Accuracy = 58.2

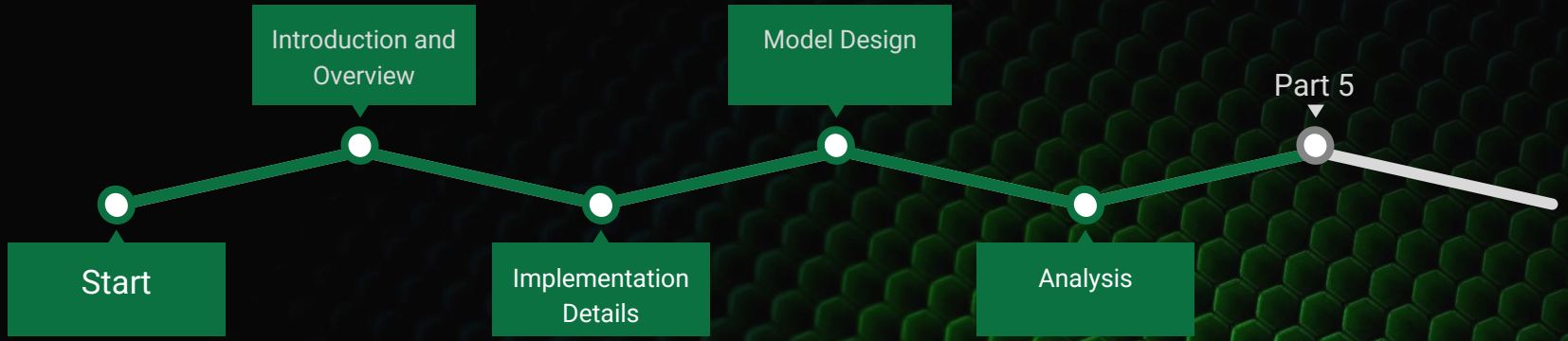
Character Confusion Analysis

- It is getting confused between some lowercase and uppercase letters like (z,Z) or (y,Y) or (x,X) or (w,W) or (v,V) etc.
- The receptive fields used in the CNN is causing overlapping of the sequences and prediction error.

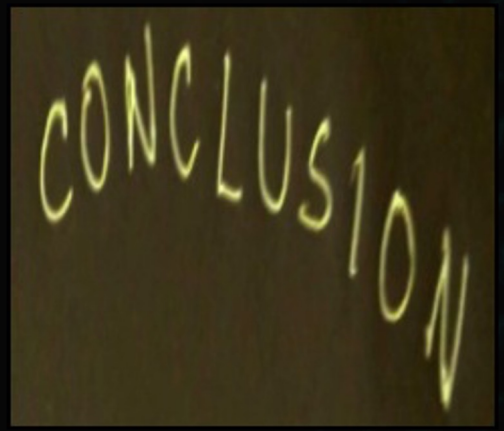
Examples:

  - N is predicted as NV or W is predicted as VW or VV.
  - M is predicted as IVI.

- The highest length of predicted word for one character set is 5 and all are against the character M.
- M is the most wrongly predicted character in terms of extra word length

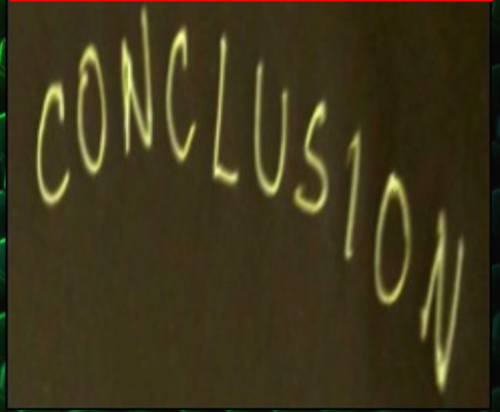


Conclusion and Future Work



Voilà !!

CONCLUSION



Future Work

- ➔ Train for longer and with a bigger dataset
- ➔ Train with irregular or augmented images
- ➔ A extension of this project would be training with a very small lexicon of commonly used English words and giving them some bias for a better prediction
- ➔ Explore the of Bidirectional Transformers in Semantic Recognition Stage

References

- Shi, Baoguang, Xiang Bai, and Cong Yao. "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition." IEEE transactions on pattern analysis and machine intelligence39.11 (2016): 2298-2304.
- Liao, Minghui, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. "Textboxes: A fast text detector with a single deep neural network." In Proceedings of the AAAI conference on artificial intelligence, vol. 31, no. 1. 2017.
- Jaderberg, Max, et al. "Spatial transformer networks." arXiv preprint arXiv:1506.02025 (2015).
- Chen, Yuxin, and Yunxue Shao. "Scene Text Recognition Based on Deep Learning: A Brief Survey." 2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN). IEEE, 2019.
- Bleeker, Maurits, and Maarten de Rijke. "Bidirectional scene text recognition with a single decoder." arXiv preprint arXiv:1912.03656 (2019).
- Yu, Deli, et al. "Towards accurate scene text recognition with semantic reasoning networks." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

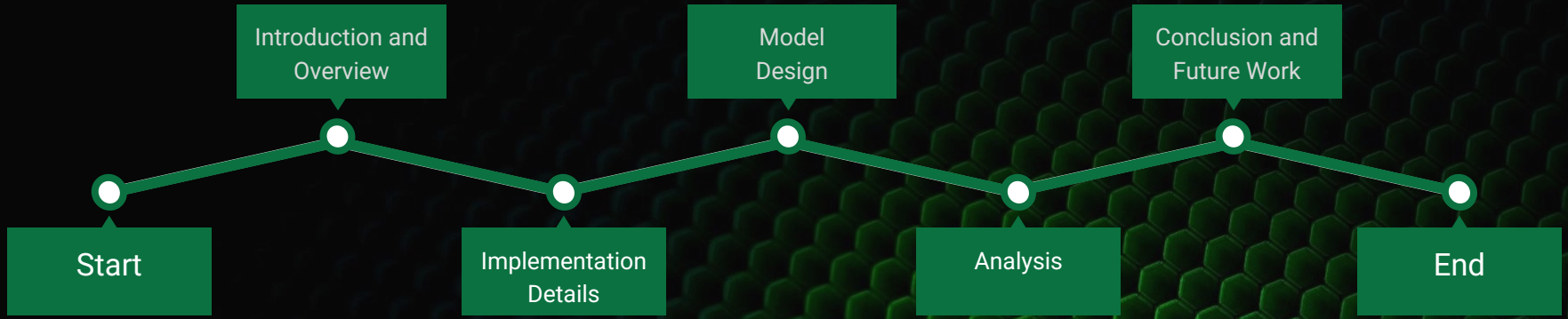
- Shi, Baoguang, et al. "Robust scene text recognition with automatic rectification." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
- Baek, Jeonghun, et al. "What is wrong with scene text recognition model comparisons? dataset and model analysis." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.
- Cheng, Zhanzhan, Yangliu Xu, Fan Bai, Yi Niu, Shiliang Pu, and Shuigeng Zhou. "Aon: Towards arbitrarily-oriented text recognition." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5571-5579. 2018.
- Liu, Zichuan, Guosheng Lin, Sheng Yang, Jiashi Feng, Weisi Lin, and Wang Ling Goh. "Learning markov clustering networks for scene text detection." arXiv preprint arXiv:1805.08365 (2018).
- Jian, Qishu. "Scene Text Detection Using Context-Aware Pyramid Feature Extraction." In 2020 International Conference on Computing and Data Science (CDS), pp. 226-230. IEEE, 2020.
- Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9, no. 8 (1997): 1735-1780.
- Kolchinsky, Artemy, and David H. Wolpert. "Semantic information, autonomous agency and non-equilibrium statistical physics." Interface Focus 8, no. 6 (2018): 20180041.

- X. Liu, G. Zhou, R. Zhang and X. Wei, "An Accurate Segmentation-Based Scene Text Detector with Context Attention and Repulsive Text Border," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 2344-2352, doi: 10.1109/CVPRW50498.2020.00283.
- T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao and C. Sun, "An End-to-End TextSpotter with Explicit Alignment and Attention," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 5020-5029, doi: 10.1109/CVPR.2018.00527.
- Zacharias, Ebin & Teuchler, Martin & Bernier, Bénédicte. (2020). Image Processing Based Scene-Text Detection and Recognition with Tesseract.

Link for other References:
https://docs.google.com/document/d/1MPLuLJIvsWqiNipgiooo68QArEtMNJU_SYUGJq68KKI/edit?usp=sharing

Authors

- Aditya Amrit (SC18B087)
- Asish kumar Mishra (SC18B074)
- Sri Aditya Deevi (SC18B080)
- Kothadiya Princekumar Balkrushna (SC18B078)


Introduction and Overview
Model Design
Conclusion and Future Work
Start
Implementation Details
Analysis
End

Thank You !