AcTiVComp20

EffDB-Unet: Superimposed Text Detection System for Arabic text

Lokesh Nandanwar
lokeshnandanwar150@gmail.com

Dr. P. Shivakumara
shiva@um.edu.my

Prof. Umapada Pal
umapada@isical.ac.in

Dr. Ramachandra Raghavendra
raghavendra.ramachandra@ntnu.no

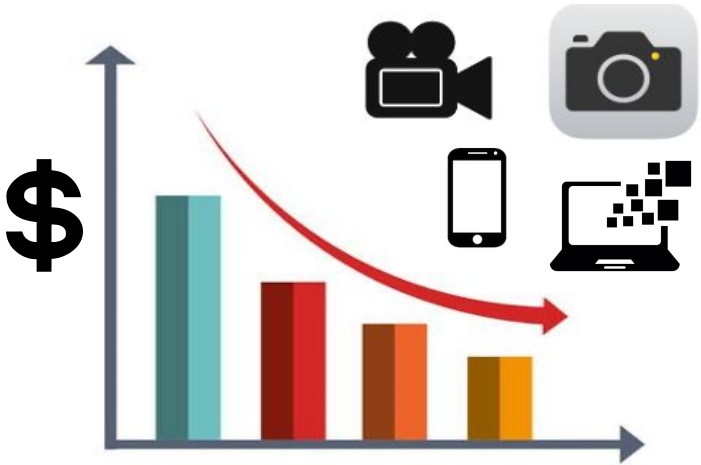
About Speaker

Lokesh Nandanwar

LinkedIn: lokeshkvn
Github: lokeshkvn
Web: lokeshkvn.github.io

- Current :
  - Project-Linked-Person under **Prof. Umapada Pal**, ISI Kolkata, India
  - M.Sc. Student (Computer Vision and Image Processing), University of Malaya, Malaysia
- Belt and Road Award 2019, China Students Service Outsourcing Innovation and Entrepreneurship Competition
- Winner - Smart India Hackathon 2019
- Student Developer, Google Summer of Code 2018
- 1 Journal Publication in **Expert Systems with Applications 2020** and 2 in **IJPRAI 2020**
- 6 Conference Publications (2 in **ICPR 2020**, 2 in **ICPRAI 2020**, 1 in **DAS 2020**, 1 in **ICACCP 2019**).

Introduction



- **Low pricing of** multimedia tools and devices.
- **Cheaper and freely* available** digital data storage and access facility.

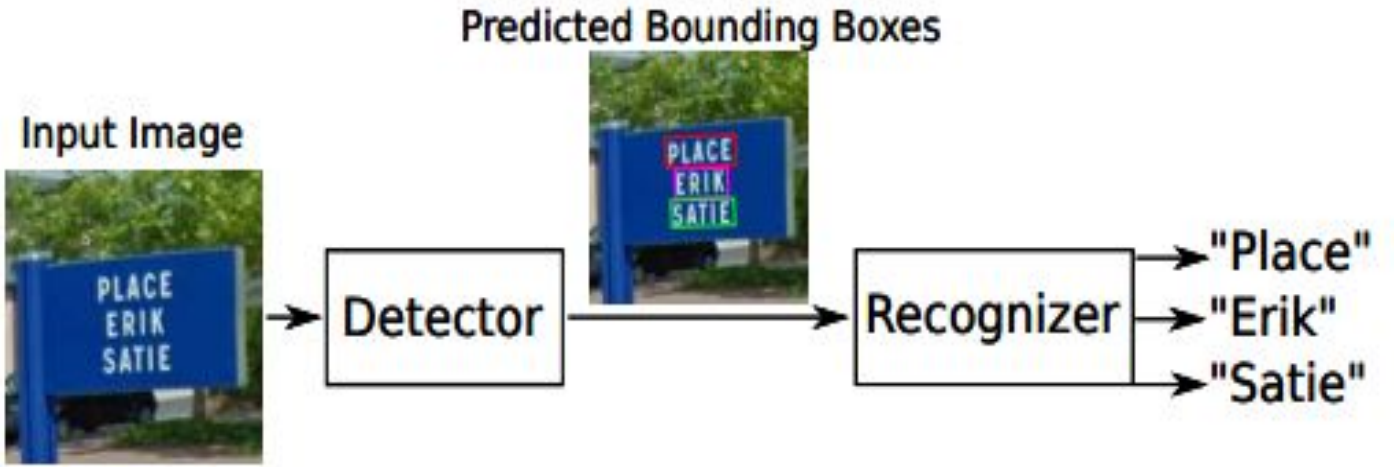
- Increase in **digital culture** and advancements.
- Increase in **digital content usage** Ex. Social media platforms.

Need to automate Text Detection/Recognition for many useful purposes!

Why Text Detection/Recognition?

- **Reading** of Scanned documents, PDFs, package labels, numbers etc.
- Annotating the images and video through **Captions**.
  - Better indexing and retrieval at semantic level.
- Automatic **number plate recognition** at toll booths
- **Street boards reading** in case of unmanned vehicles.
- Providing **scene information** to visually impaired people.
- **Events extraction** from sports, news broadcast, etc.
- **Tracing and watching** the persons using T-shirt label/Number.
  - Marathons, Exhibitions, processions, etc.

Complete flow for Digital text reading



Our Focus: Text Detection

- Process of **detecting** the text present in the image, followed by surrounding it with a rectangular bounding box.
- The image is **segmented** into multiple segments of texts.
- Each segment is a **connected component of pixels** with similar characteristics (Characters).

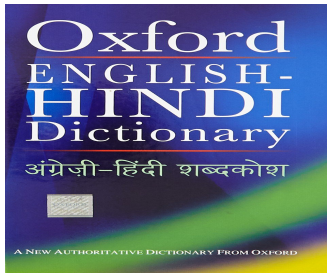
General Challenges

Uneven Lighting

Scene Complexity

Uneven Color

Blurring/degradation



Multilingual

Arbitrary Shaped text

Objectives

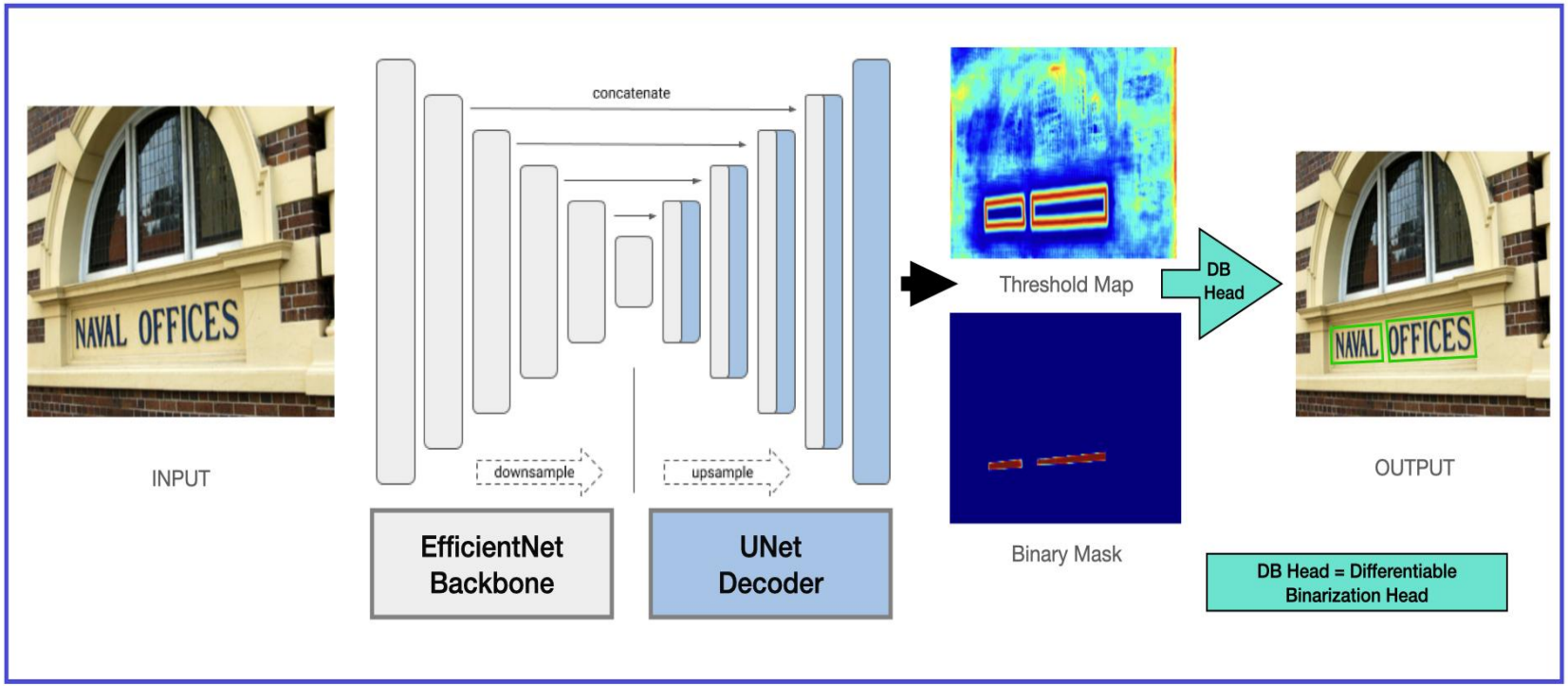
- So far, the methods have **focused only on some languages** such as English, Latin and Chinese.
- For a language like **Arabic** which is also used by more than one billion people around the world, the literature is limited to very few studies.
- This presentation aims to tackle the challenges in **Arabic Text Detection** in News Video Frames.

Existing Methods and Drawbacks

Some Recent State of the Art Methods

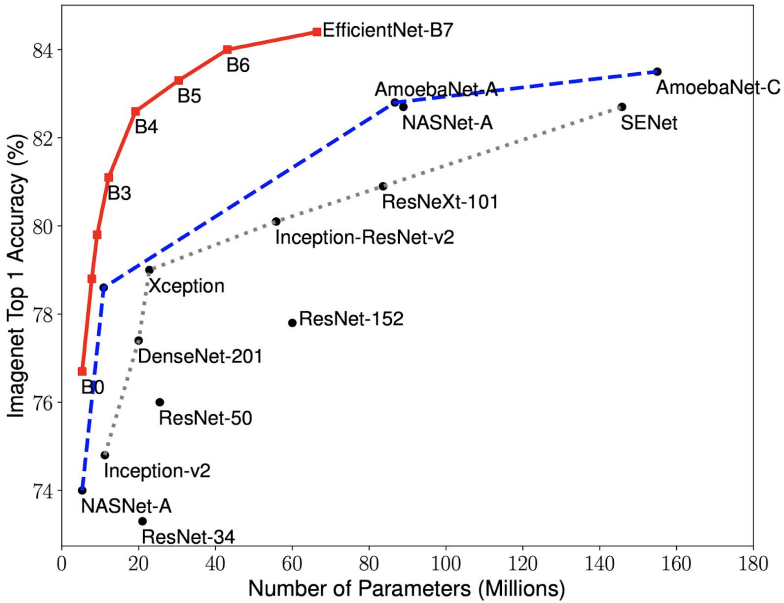
1) **FOTS (Fast oriented text spotting)** : Weak backbone (Resnet) and no segmentation head [CVPR 2018] *(https://arxiv.org/abs/1801.01671)*
2) **PSENet(Progressive Scale Expansion Network)** : Weak Backbone and weak decoder [CVPR 2019] *(https://arxiv.org/abs/1806.02559)*
3) **CRAFT(Character Region Awareness for Text detection)** : Weak feature extractor (VGG-Unet) [CVPR 2019] *(https://arxiv.org/abs/1904.01941)*
4) **DB-Net (Differential Binarization Network)** : Weak backbone (Resnet) [AAAI 2020] *(https://arxiv.org/abs/1911.08947)*

Proposed Method (EffDB-UNet)



Why EfficientNet as Backbone?

- Designed through **Neural Architecture Search (NAS)**
- Based on Model scaling method using **effective compound coefficient.**
- Uses low Parameters and High Accuracy, implies **less training time** and improve in real time performance.
- Superpasses state-of-the-art accuracy with up to **10x better efficiency (smaller and faster).**



Reference: https://arxiv.org/abs/1905.11946

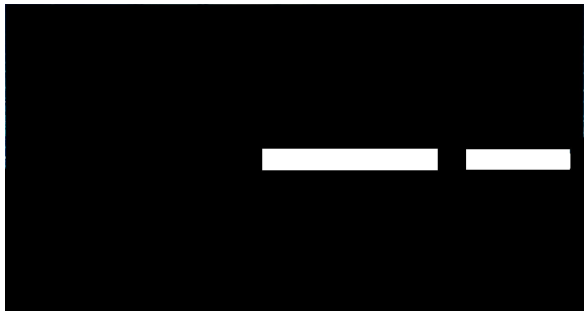
Why UNet as Decoder?

- Initially designed for **medical image** segmentation, then adapted for **fast and precise segmentation** in most of the Computer Vision Tasks
- Most **Reliable and cost-effective** decoder for segmentation tasks.
- **Performs better or nearly same** than even the most recent architectures such as DeepLab, FPN, FCN, PSPNet, etc
- Needs **less parameters**, effectively reducing training and inference time.

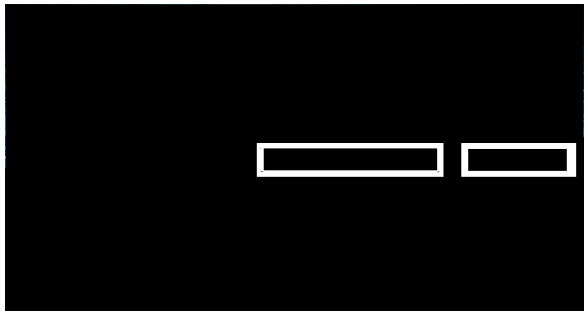
Reference: https://arxiv.org/abs/1505.04597

Binary Mask with Threshold Map!!!

Input Image



Binary Mask



Threshold(Border) Map

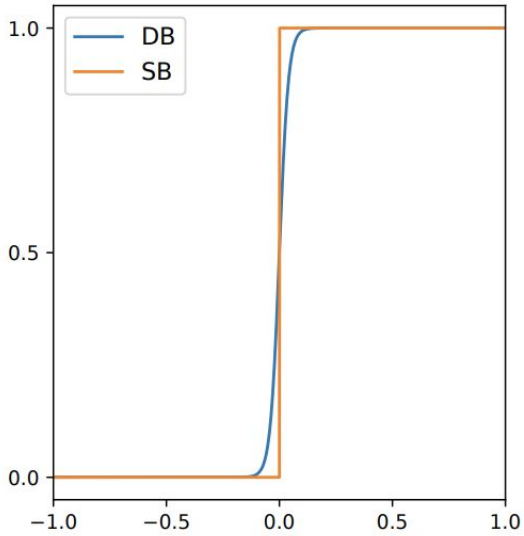
- Optimal strategy for Segmentation of words in case of **text lines/ crowded text (Superimposed segmentation in general).**
- Precise and accurate word wise segmentation helps in **effective recognition.**

DB-Head

- **Differential Binarization (DB)** instead of Standard Binarization (SB).
- The major effect of DB-Head is differentiability, which makes the process of **binarization end-to-end trainable** in a CNN.

$$\hat{B}_{i,j} = \frac{1}{1 + e^{-k(P_{i,j} - T_{i,j})}}$$

- The differentiable binarization with adaptive thresholds help to differentiate text regions from the background and also to **separate text instances which are closely jointed.**



Reference: https://arxiv.org/abs/1911.08947

Quantitative Results

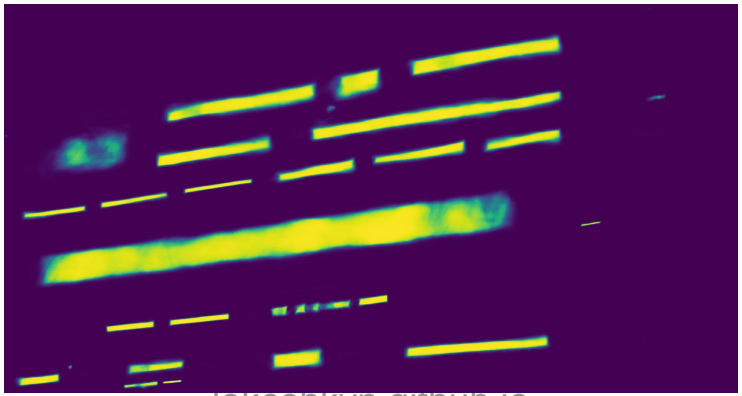
- Results on Complete Public test dataset from competition:

Precision	99.4378 (513 images)
Recall	91.8398 (513 images)
F-measure	95.4879

Reference: https://diuf.unifr.ch/main/diva/AcTiVComp/

Qualitative Results

Results on Multi-lingual Images



Frameworks and Codes

- DL Framework: Pytorch using Catalyst
- Encoder and Decoder: Segmentation Models Pytorch
- Segmentation Head (DB-Head): Real-time Scene Text Detection with Differentiable Binarization

Explore and Cite our works

- 3DTDS: 3D Video Text Detection System (**Under Revision**, will be available soon)
- Forged IMEI Numbers and Air Ticket Detection (**ESWA 2020**)
- Classification of 2D & 3D Texts in Video/Scene Images (**ICPR 2020, IAPR DAS 2020**)
- Deep CNNs for Detecting Forged Handwriting (**ICPR 2020**)
- Detecting Altered Text in Document Images (**ICPRAI 2020, IJPRAI 2020**)
- Caption and Scene Text Classification in Action Video Images (**ICPRAI 2020, IJPRAI 2020**)

Visit: lokeshkvn.github.io for more details

Thank You!