SIW 2021: ICDAR Competition on Script Identification in the Wild

Abhijit Das[1],[2], Miguel A. Ferrer[3], Aythami Morales[4], Moises Diaz[5], Umapada Pal[1], Donato Impedovo[6], Hongliang Li[7], Wentao Yang[7], Kensho Ota[8], Tadahito Yao[8] Le Quang Hung[9], Nguyen Quoc Cuong[9], Seungjae Kim[10], and Abdeljalil Gattal[11]

[1] Indian Statistical Institute, Kolkata abhijit.das@thapar.edu
[2] Thapar University, India
[3] Univ. de Las Palmas de Gran Canaria, Spain
[4] Universidad Autonoma de Madrid, Spain aythami.morales@uam.es
[5] Universidad del Atlántico Medio, Spain moises.diaz@atlanticomedio.es
[6] Università degli Studi di Bari Aldo Moro donato.impedovo@uniba.it
[7] South China University of Technology, China
[8] Canon IT Solutions Inc., Japan
[9] University of Information Technology, Vietnan
[10] NAVER Papago, Korea
[11] Larbi Tebessi University, Argelia

Abstract. The paper presents a summary of the 1st Competition on Script Identification in the Wild (SIW 2021) organised in conjunction with 16th International Conference on Document Analysis and Recognition (ICDAR 2021). The goal of SIW is to evaluate the limits of script identification approaches through a large scale in the wild database including 13 scripts (MDIW-13 dataset) and two different scenarios (handwritten and printed). The competition includes the evaluation over three different tasks depending of the nature of the data used for training and testing. Nineteen research groups registered for SIW 2021, out of which 6 teams from both academia and industry took part in the final round and submitted a total of 166 algorithms for scoring. Submissions included a wide variety of deep-learning solutions as well as approaches based on standard image processing techniques. The performance achieved by the participants prove the elevate accuracy of deep learning methods in comparison with traditional statistical approaches. The best approach obtained classification accuracies of 99% in all three tasks with experiments over more than 50K test samples. The results suggest that there is still room for improvements, specially over handwritten samples and specific scripts.

Keywords: Handwritten and printed script identification · wild · deep learning · multi-script.

1 Introduction

Due to the ever-increasing demand for the creation of a digital world, many Optical Character Recognition (OCR) algorithms have been developed over the years [1]. Incidentally, script identification plays a vital role in OCR pipeline. It use is also been used for several application such as signature verification [2, 3], [4], scene text detection [5], [6]. A script can be defined as the graphic form of the writing system used to write a statement [7], [8].

The availability of large numbers of scripts makes the development of a universal OCR a challenging task. This is because the features needed for character recognition are usually a function of structural script properties and of the number of possible classes or characters. The extremely high number of available scripts makes the task quite daunting and sometimes deterring, and as a result, most OCR systems are script-dependent [9]. The approach for handling documents in a multi-script environment is divided into two steps: first, the script of the document, block, line or word is estimated, and secondly, the appropriate OCR is used. This approach requires a script identifier and a bank of OCRs, at a rate of one OCR per possible script. Many script identification algorithms have been proposed in the literature. Script identification can be conducted either offline, from scanned documents, or online, if the writing sequence is available [10].

Identification can also be classified either as printed or handwritten, with the latter being the more challenging. Script identification can be performed at different levels: page or document, paragraph, block, line, word, and character. As it is similar to any classical classification problem, the script identification problem is a function of the number of possible classes or scripts to be detected. Furthermore, any similarity in the structure of scripts represents an added challenge [7].

Hence, to elevate state-of-the-art several benchmarking effort by publishing publicly available datasets [11, 12] and competition has been organised [13–17]. Consequently, the benchmarking works on script identification in the literature uses different datasets with different script combinations . Therefore, it is difficult to carry out a fair comparison of these different approaches. Moreover, the databases employed in related studies usually include two to four scripts. A few actually include an even higher number of scripts but with not exhaustive combination with both handwritten and printed samples with different level of annotation (word, line and document).

Hence to alleviate this drawback, in this competition we aim to offer a database for script identification, which consists of a wide variety of some the most commonly used scripts, collected from real-life printed and handwritten documents. The competition is also aim to document the recent development in this area of research and attract the attention of the researchers. Specifically, we aim to answer the following questions:

– How do contemporary script identification techniques perform with large scale challenging document images captured in the wild?

- What impact do changes in type of data (handwritten and printed) have on identification performance?

The following contributions that are documented in this report:

- A rigorous evaluation of several contemporary script identification approaches.
- A comprehensive analysis of the identification approaches.
- A public benchmark with more 80K images from 13 scripts obtained from real handwritten and printed documents.

2 Benchmarking Dataset

We developed a large dataset for script identification tasks, consisting of printed and handwriting documents of the following 13 scripts: Arabic, Bengali, Gujarati, Gurmukhi, Devanagari, Japanese, Kannada, Malayalam, Oriya, Roman, Tamil, Telugu, and Thai [18]. Figure 1 shows an example of words in each script.

The printed documents were collected from local newspapers and magazines, whereas mostly native volunteers provided the handwritten. All documents were scanned at 300 dpi of resolution. However, various conditions are included in the databases, like different inks, sheets, font sizes, and styles. As a consequence, controlled background removal and ink equalization was applied to ensure a cleaner database. Furthermore, the word segmentation from the documents was carried out by an automatic system, which was manually fine-tuned later by checking all individual words.

This novel dataset has not shared at the time of the competition. In addition, we only consider words extracted from the texts for this competition. Specifically, the word-based data was divided into *Training* and *Testing* sets for this competition, as summarize Table 1.

- *Training.* All registered participants in the competition had access to the training data. The more than 30K training images were divided into two main subsets with printed (21974) and handwritten (8887) images.
- *Testing.* Once the deadline was reached, we shared the testing data with the participants, which consisted of 55814 unlabelled images to identify in the 13 scripts. The test set includes more than 50K handwritten and printed images. The type of data (handwritten or printed) as well as the script label, were not provided to the participant until the end of the competition.

As can be seen, the number of words is different in each script. It makes the script identification an unbalance challenge with this benchmarking corpus. Finally, the images within *Training* and *Testing* sets were not the same. They were randomly extracted from the words available in the MDIW-13 multiscript document database [19]. In other words, the data in this competition is a subset of the large database. The database is publicly available[12].

[12] https://https://gpds.ulpgc.es/

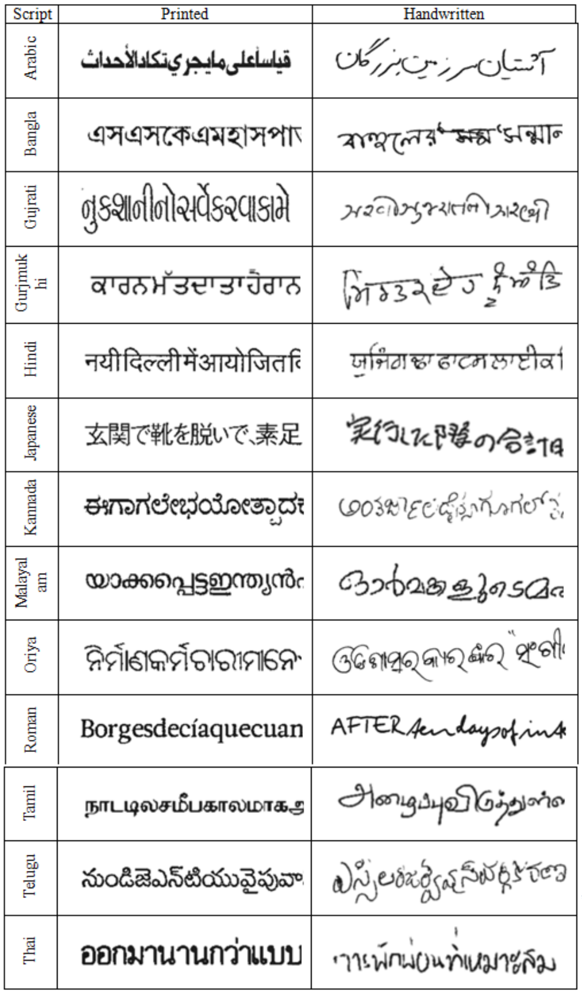

Script
Printed
Handwritten
Arabic
Bangla
Gujrati
Gurjmukhi
Hindi
Japanese
Kannada
Malayalam
Oriya
Roman
Borgesdecíaquecuan
AFTERtendaysofink
Tamil
Telugu
Thai

Fig. 1. Example of image-based words used in the competition.

Table 1. Summary of the word images included in each script for the *Training* and *Testing* sets.

Script	*Training*		*Testing*	
	Printed	*Handwritten*	*Printed*	*Handwritten*
Arabic	1996	570	4206	3370
Bengali	1608	401	949	8919
Gujarati	1229	144	982	37
Gurmukhi	3629	538	5475	135
Devanagari	1706	1203	1076	301
Japanese	1451	352	363	89
Kannada	1183	872	974	1123
Malayalam	2370	575	1950	144
Oriya	1660	333	649	7514
Roman	1574	558	6053	3750
Tamil	451	873	1667	557
Telugu	1261	640	865	161
Thai	1856	1828	1861	2644
Total	21974	8887	27070	28744

3 Evaluation protocol

SIW 2021 was executed in two stages. During the first stage participants were given the training split of the MDIW-13 datset, including the ground truth and were asked to develop their algorithm. In the second stage, the test split of the MDIW-13 (without the annotation) was provided to the participants to infer the script label on the test images.

The detailed tasks for the competition are as follows: 1) **Task 1:** Script identification in handwritten document; 2) **Task 2:** Script identification in printed document; and 3) **Task 3:** Mixed script identification: Train and tested with handwritten and printed.

The evaluation measure used during the competition was the **Correct Classification Accuracy**. This performance measure was calculated as the percentage of samples correctly classified respect the total number of samples available for each of the tasks. Note that training and test sets present certain class imbalance and the methods need to deal with this challenge.

Participants performed word level script recognition. The submission which achieved the best Correct Classification Accuracy for Task 3 was considered as the winner. The ground truth was manually annotated and segmented according to a semiautomatic process described in [19].

4 Details of submitted approaches

Six different groups submitted their approaches for the final evaluation. The participants include teams from academia and industry. Table 2 presents a summary of the participating groups and their approaches. As can be seen, the proposed

Table 2. Summary of participants and submitted approaches to SIW 2021. The table lists the abbreviations of the models, as used in the experimental section. PR = Pre-trained models, EX = External data, HC = Hand-crafted features, AL = Detection and alignment, EM = Ensemble models, DM = Differentiate models, Pre = Pre-processing, post = Post-processing, ✓= Yes, x = No.

No.	Group	PR	EX	HF	AL	EM	DM	Pre	Post
1	Ambilight	✓	✓	x	✓	x	x	✓	✓
2	DLVC-Lab	✓	✓	x	x	✓	x	x	x
3	NAVER Papago	✓	x	x	x	x	x	✓	x
4	UIT MMlab	✓	x	x	x	✓	✓	x	✓
5	CITS	✓	✓	x	x	x	x	✓	✓
6	Larbi Tebessi	x	x	✓	x	x	x	x	x

approaches show heterogeneous characteristics: with and without pre-processing or post-processing techniques, ensemble or unique models, use of augmented data. We proceed to present a summary of the best systems submitted by each of the participants.

4.1 The Lab of Ambilight, NetEase Inc. (Ambilight)

This team used semantic segmentation method as our baseline model instead of classification method. The semantic segmentation model is more focused on the details of every character and can reduce the disturbances of background, so the semantic segmentation model is better in this task. To fully utilize the classification label, a multi-task training design is introduced to further improve the performance of the segmentation task. Therefore, a classification branch is added in our proposed framework. Also, to fit text geometric features better, attention module and deformable convolution are added into the backbone. Another highlight of our approach lies in that we use lots of synthetic data and grid distortion technique to simulate the handwriting style of different people, which are finally proved valid tricks in this task. During testing phase, we apply semi-supervised learning technique to fit the test data better. All these strategies stated above make us achieve the top performance in the competition.

Introduction and Motivation: As stated above the challenges such as variation in length of texts, ever-changing division, and similar letters even characters existing in different scripts, hence an explicit solution is to design a framework based on the fine-grained classification work. Currently, a popular model of fine-grained classification is mainly based on the attention mechanism, such as the WS-DAN[20] model (proposed by MARA in 2019), where it uses the attention mechanism to crop the partial details of the image to assist the classification. However, its attention mechanism is actually a weak supervision mechanism. For detection-based models, they generally have to use stronger supervision information. For example, for the Part-RCNN[21], the foreground is first detected, and the detected foreground is scored with a discrimination degree. More detailed supervision information is based on segmentation methods.

Table 3. Some insight result on different architecture by team Ambilight.

Model	Printed (mIoU)	Handwritten (mIoU)
VGG19	0.9012(Acc)	-
ResNet50-FCN	0.9590	0.8327
ResNest50-DeepLab3+	0.9704	0.864
HRnet48-OCRnet	0.9732	0.8732
HRnet48-OCRnet+DCN	**0.9795**	**0.8958**
Swin-Transfromer	0.9654	0.8768

Fine-grained backgrounds are often complex and different, but the foregrounds are very similar. If the foreground can be segmented for classification, better performance will be achieved. Similarly, Mask-CNN[22] is a fine-grained classification model based on strongly supervised segmentation information, but due to the high cost of labelling, it is rarely applied in the industry.

Although only the classification information is given in this competition, the image only filled with black characters on a white background allows us to calculate the mask required for segmentation directly through the pixel information. Later, the participants compared different encoder and decoder combinations, tried the classic VGG[23], Resnet[24] as the encoder, FCN[25], Deeplab3+[26] as the decoder. Also tried the newly proposed Resnest[27], Swin-Transfromer[28], etc.In the end, we combine the current SOTA backbone HRnet[29] and segmentation decoder OCRnet[30] to solve this task. In addition, they also replace part of the normal convolution in the encoder for DCN[31] convolution to play a role of weak attention supervision. The following table demonstrates their comparative study on different methods and backbones.

Detailed Method Description: (a) Since the training data is black on a white background, the mask required for segmentation can be obtained according to the pixel values. After attained the image mask, each pixel has supervision information, and more local features can be extracted for very similar languages. Here the participant use HRnet as the encoder to extract image features, and then use OCRnet as the decoder for the output the segmentation results, and determine the final output category according to the segmentation vote of each pixel. In addition, they additionally designed an auxiliary classifier for training to make the model also pay attention to the global features, and used DCN v2 (Improved Deep Cross Network) convolution instead of 2D convolution to strengthen the modeling ability of text shapes.

(b) Pseudo-label Fine-tuning [32]: The team use the trained model to do inference on test set. If more than 70% of the pixels (foreground pixels) are classified to be the same language, they assume that the predication is correct. Samples that meet this condition will be allocated to the training set for fine-tuning. In this way, iteratively fine-tune the model in multiple rounds.

(c) Synthetic Data and Data Augmentation: For the printed scripts, they generate millions of synthetic data in different scripts and fonts using text renderer, which can greatly expand the training data. Firstly, they use the synthetic data to pre-train the model, and then use the given training set for finetuning. For the handwritten scripts, they use grid distortion to simulate handwriting changes. This enhancement method can appropriately simulate non-rigid deformations such as changes in the thickness and length of human strokes to enhance the robustness of model for different handwriting.

(d) Loss Function: The team did a lot of experiments on the loss function, and finally used focal loss [33] to increase the learning weight of the text part.

(e) Augmentation during Testing: They use random-crop-resize during training, which plays a small-scale multi-scale role. Therefore, during the inference on the testing set, multi-scale resizing is performed on the intput image. Results of different sizes are blended together to the final result.

4.2 South China University of Technology (DLCV-Lab)

The three tasks were treated as classification problems and were solved by adopting deep learning methods. In order to improve the diversity of training data, the team utilized data augmentation technique [34] and synthesized a dataset using fonts in different scripts. Finally, they ensemble three CNN-based models, namely, ResNet-101 [24], ResNeSt-200 [27] and DenseNet-121 [35] with CBAM [36] for better classification accuracy.

The first question they considered is whether they should adopt machine learning methods or deep learning methods. In addition to more than 30,000 samples in the MDIW-13 database, they also use the collected fonts to generate a synthetic dataset. With such a relatively large amount of data for training, they thought that deep learning methods may outperform machine learning methods. Among image processing methods in deep learning, Transformer-based models has received lots of recent attention, while CNN-based models are still the mainstream. In the absence of the massive training data required by transformer, they believe CNN-based models may be more suitable. After determining the main technical route, on one hand, we try various CNN-based models and ensemble the best three models for better classification accuracy. On the other hand, they collect some fonts for synthesizing data, and utilize the data augmentation technique for text images to improve the diversity of the training data.

Synthesize Data: To synthesize data, fonts in 13 scripts are collected from the Internet, and corpora are translated from 58,000 English words using the Google translation API. Then they randomly select fonts and the translated word corpora to generate 5,000 images for each script, which are added to the training set.

Data Augmentation: The data augmentation technique they used [34] embodies different transformations for text images, including distortion, stretch and perspective. In the training phase, there is 50% probability for every sample to execute each transformation.

Details of Model: After trying a variety of models, the team choose three separately trained models, Resnet-101, Resnest-200 and Densenet-121 with CBAM to ensemble. We use the Adam optimizer with a weight decay of 1e-4 and a learning rate of 1e-4. The learning rate is set manually to 1/10 of its current value when the loss value no longer drops. The cross-enropy loss function is adopted and image resolution of input is set to 300×700. All three models are pre-trained on ImageNet. It is worth noting that we did not use the synthetic data for Resnet-101, as it will cause a worse result. Finally, the prediction confidences of the three models are averaged as the output.

4.3 NAVER Papago (NAVER Papago)

Given the time constraint of the competition, the main strategy and motivation was to fix the network and conduct quick experiments to conquer the problems of the data. As many other real world problems do, the data had class imbalance problem. Also, the images didn't have rich pixel level information such as color, contrast, but had much more spatial level information such as shape, font style. To address these issues, first of all, the participant used stratfied data sampling to overcome class imbalance. This helps the model fit to less-frequent-class images and boosts overall score. Secondly, many spatial level augmentations were applied to make the model better recognize the newly seen text shapes in test data. Augmentations such as random shift, scale, stretching, grid distortion were applied, and in accordance, same augmentations were applied at test time which also improved score. The network architecture was fixed to ResNet50 for all the experiments above, and was changed to NFnet-f3 near the end of competition. In conclusion, the overall approach was to leave most of the settings fixed and concentrate on one or two most important issues of the competition.

Preprocessing: All images were rescaled, maintaing the width/height ratio, and padded to have the same size 160x320.

Data split: 10% of each handwritten and printed data were used as validation set. They were sampled by stratified sampling, where the ratio of each class in the sample remains the same as the ratio of each class in the whole data. Then training data and validation data of both handwritten and printed data were mixed, for training all at once.

Inference: 1) pseudo-labeling: test predictions that had confidence over 0.99 were pseudo-labeled and used to finetune model before inference; 2) test time augmentation: test time augmentation with random scale, random horizontal, vertical stretching was applied.

4.4 University of Information Technology (UIT MMLab)

Our approach is building a two-stage deep learning system for script identification. In the first stage, we applied a residual neural network (ResNet) [37] to classify the script as handwriting or print. In the second stage, for each type of these we use our corresponding EfficientNet [38] model to identify the script. For the best result on private test set, we used EfficientNet-B7 for handwritten script and EfficientNet-B4 for printed script.

Handwritten/Printed type classification: A Resnet-50 architecture pre-trained on ImageNet was used as backbone network. We stacked 1 fully connected layer with 1024 units in front of the output layer. A sigmoid loss function was used as binary classification. An Adam optimizer with learning rate of 0.0001 was used for the training process. The dataset was splitted with the ratio of 9/1 and then trained in 20 epochs with a batch size of 16. We saw that 20 epochs are enough for the model to converge.

We trained separately 2 models with regard to 2 different random seeds in train-validation step. It comes to my attention that one of these 2 models helps to get higher score on printed task. The another model leads to higher score on handwritten task. So, in the inference phase, the predicting result of 2 models are compared with each other. The images that make their result different will be considered manually by a visualizing tool. Following these steps, we can create a quite significant handwritten-printed classification result.

Script identification: For script identification phase, we decided to use two separate EfficientNet B4 and B7 models for handwritten and printed scripts. The backbone network we used is a pretrained network previously trained on a large ImageNet dataset contains 1000 classes labels, we can take advantage of pretrained weights to extract useful image features without retraining from scratch. With transfer learning approach, we exclude the final fully connected (FC) layer of pretrained model, then we replaced the top layer with custom layers containing FC layers for identify 13 languages of script allows using EfficientNet as a feature extractor in a transfer learning workflow. The features is fed into global average pooling (GAP) to generate a vector whose dimension is the depth of the feature, this vector is the input of the next FC Layer. The GAP layer outputs the mean of each feature map, this drops any remaining spatial information, which is fine because there was not much spatial information left at that point. The final FC layer 1×13 using softmax activation produces the probability of each class ranged from 0 to 1.

Because freezing EfficientNet and training only custom top layers tends to underfit the training data, training both EfficientNet and custom top layers tends to overfit the training data, so the approach is freezing some first layers of EfficientNet to make use of the low level features extracted by pretrained network on ImageNet datasets, then training the remaining layers and top layers. The team use Adam optimizer with a learning rate of 0.0001 to minimize the categorical cross entropy loss function. For each type of script (handwritting and print), They splitted the data into training set and validation set with the ratio of 8/2. Then, they trained two base models i.e B4, B7, 250 epochs for each model with a batch size of 16 and only save the best weight with the highest validation accuracy.

4.5 Canon IT Solutions Inc. (CITS)

Firstly, the participants from this groups made patches by sliding window. Stride was 56pixels[13]. Secondly, they classified each patch using a Efficient Net[39] as a classifier.

[13] 56 is 224/4, and 224 is CNN's default input width

Lastly the participants calculated sum of confidence of each class of all patches, and adopted the class corresponding to maximum argument as the inference result.

For prepossessing, we used shave 20 pixels(up, bottom, left, right), resize height to 224 keeping aspect ratio, normalization. The participants used Tesseract in post-processing for printed Hindi's and Gurmukhi's results.

There were 3 postprocessing steps: 1) trained a CNN model that classifies images are printed or handwritten; 2) collected images which were classified as printed Hindi or printed Gurmukhi; 3) OCR'ed each image in step 2 with Tesseract of Hindi model and Gurmukhi model. Tesseract outputs confidence score with OCR result, and we adopted script with higher confidence score.

The participants generated pseudo handwritten images with CycleGAN [40] and used as a training dataset. This dataset contained 13,000 images(1,000 images for each class).

4.6 Larbi Tebessi University (Larbi Tebessi)

Among the different methods for research purposes, the texture-based descriptor is preferred by many researchers due to its various advantages, strengths and benefits. Proposed research method allows to extract highly informative elements of the printed and handwritten text. Otherwise, they do not involve comparing pieces of text like-for-like. In this way, the computational cost provides good insight into the complexity of the system compared to other systems. However, the proposed research method also has their weak point that must be considered which is sensitive to noise that makes the extracted features sensitive to small changes in the handwriting. The textural information is captured using an oriented Basic Image Feature (oBIF) columns.

In order to increase the performance of the oBIFs descriptor, the participants combine oBIFs at two different scales to produce the oBIF column features by ignoring the symmetry type flat. The oBIFs column features are generated using different values of the scale parameter σ while the parameter ϵ is fixed to 0.001. The generated feature vector is finally normalized.

The oBIFs column histograms are extracted, the both oBIFs column histograms for the scale parameter combination (2,4) and (2,8) are concatenated together to form the feature vector representing each printed and handwritten image. Once the features are extracted, classification is carried out using Support Vector Machine (SVM) classifier. We have employed the Radial Basis Function (RBF) kernel with the kernel parameter selected to 52 while the soft margin parameter C is fixed to 10. The participants evaluated the oBIF column features to identify scripts from printed and handwritten images on the dataset of the ICDAR 2021 competition on Script Identification in the Wild (SIW 2021). The experiment is carried out by using both printed and handwritten samples in the training and test sets.

Table 4. Summary of final results for each of the three tasks. Correct classification accuracy (final rank in brackets). The table presents two baseline methods: 1) the Dense Multi-Block Linear Binary Pattern [19] and 2) the Random Chance. T.# Subm = Total submissions

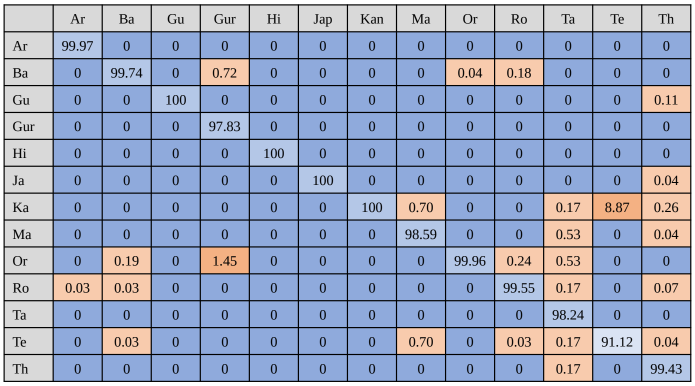
Group	# Subm	Task 1 (Handwritten)	Task 2 (Printed)	Task 3 (Mixed)
Ambilight	16	99.69% (1)	99.99% (1)	99.84% (1)
DLVC-Lab	43	97.80% (3)	99.80% (2)	98.87% (2)
NAVER Papago	26	99.14% (2)	95.06% (5)	97.17% (3)
UIT MMlab	46	95.85% (4)	98.63% (4)	97.09% (4)
CITS	34	90.59% (5)	99.24% (3)	94.79% (5)
Baseline [19]	-	89.78% (-)	95.51% (-)	94.45% (-)
Larbi Tebessi	1	81.21% (6)	86.62% (6)	83.83% (6)
Random	-	7.14% (-)	7.43% (-)	7.22% (-)

5 Benchmarking results with analysis and discussion

In this section we proceed to report and analysis the results obtained from the submission. The Table 4 shows the final rank of the competition for the three different tasks. The results are reported in terms of correct classification accuracy. In order to compare the improvement provided by the competition we released two benchmarking. Firstly, the random chance, which is around 7% for all three tasks. Further, we provided a second benchmark, i.e. the results obtained using the Dense Multi-Block Linear Binary Pattern as per the reported in [19].

We can observe from the Table 4 that the results of the participants ranged from 83.83% to 99.84%. Most of the submission were based on CNN, and they outperformed the baseline [19]. As it is expected, the handwritten task represent a bigger challenge for the participants with accuracy's lower than in the printed scenario for most of the scenarios. Further, for the mixed task the accuracy's are someway between their printed and handwritten results. Incidentally, the submission from Ambilight i.e. the winner team had a marginal difference of .2% between printed and handwritten task. Also the gap is more less i.e. 0.06% while considering mixed and printed.The reason behind this expected to be the approach they chooses while solving this problem. They considered this problem as a semantic segmenting task and which helps to give this leverage while finding the details of the character.

We proceed to further analyse the system submitted by the winning team. The Figure 2 shows the confusion matrix obtained by the winning approach (Task 1: Handwritten script identification i.e the most challenging task). The results show that there is room for improvement in some of the scripts, specially Gujarati, Telugu, Tamil, and Malayalam. Incidentally, Gujarati script were mostly mis-classified as Bengali and Oriya, we assume that it is mostly due to the writes writing style. As a fact Bengali and Oriya script do not have much similar outlook to Gujarati. While considering Telugu the biggest mis-classification was with Kannada script nearly 8.8%, it can be considered due to the visual simi-



	Ar	Ba	Gu	Gur	Hi	Jap	Kan	Ma	Or	Ro	Ta	Te	Th
Ar	99.97	0	0	0	0	0	0	0	0	0	0	0	0
Ba	0	99.74	0	0.72	0	0	0	0	0.04	0.18	0	0	0
Gu	0	0	100	0	0	0	0	0	0	0	0	0	0.11
Gur	0	0	0	97.83	0	0	0	0	0	0	0	0	0
Hi	0	0	0	0	100	0	0	0	0	0	0	0	0
Ja	0	0	0	0	0	100	0	0	0	0	0	0	0.04
Ka	0	0	0	0	0	0	100	0.70	0	0	0.17	8.87	0.26
Ma	0	0	0	0	0	0	0	98.59	0	0	0.53	0	0.04
Or	0	0.19	0	1.45	0	0	0	0	99.96	0.24	0.53	0	0
Ro	0.03	0.03	0	0	0	0	0	0	0	99.55	0.17	0	0.07
Ta	0	0	0	0	0	0	0	0	0	0	98.24	0	0
Te	0	0.03	0	0	0	0	0	0.70	0	0.03	0.17	91.12	0.04
Th	0	0	0	0	0	0	0	0	0	0	0.17	0	99.43

Fig. 2. Confusion Matrix (Task 1) obtained by the best method (Ambilight).

larity of the two script. This was the highest mis-classification and as a reason Telugu script classification attend the lowest performance. Although it is interesting to find that Kannada script had perfect classification. Similar to Kannada and Telugu, Malayalam and Tamil script have high visual similarity, considering Tamil the highest mis-classification was with Malayalam script, but which is not the case while considering Malayalam highest mis-classification was found between Kannada and Telugu. Hence, we cannot conclude that visual pattern and structure only responsible while considering script identification multi-script scenario.

Summarising the competition, from Fig. 3 we can conclude that the maximum correct classification accuracy obtained by all the participants during the 14 days is nearly 100%. The results obtained during the first week of the competition represent a period where participants adapted their systems to the test set. We can see a performance improvement from 91% to 97.4% during this first week. The second week shows a constant improvement with a final correct classification accuracy of 99.84%. In a competition with two weeks available for submissions, we cannot discard certain overfitting to the test set. However, the large number of samples (more than 50,000) and the different characteristics of the tasks (handwritten vs printed) is a added challenge.

6 Conclusions

The 1st edition of the Script Identification in the Wild Competition was organised to evaluate and benchmark the performance of contemporary script identification techniques captured in the wild and explore the robustness of existing models *w.r.t.* to changes in the font, size, ink and printing quality used for document development and image acquisition as well as changes in the external acquisition conditions. A total of 6 groups participated in the competition and contributed 6 algorithm for the group evaluation. The identification algorithms

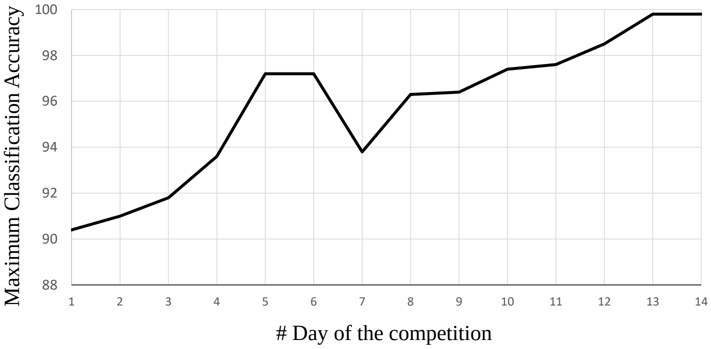


Fig. 3. Maximum correct classification accuracy (Task 3) obtained by all the participants in each day of the competition.

were compared in terms of printed, handwritten and mixed documents. Most of the submitted models ensured solid identification results in most experimental scenarios. It is worth mentioning that for some the combination of script the performance was slightly lower which requires further attention.

References

1. Line Eikvil. Optical character recognition. *citeseer. ist. psu. edu/142042. html*, 26, 1993.
2. Abhijit Das and *et al.*,. Multi-script versus single-script scenarios in automatic off-line signature verification. *IET biometrics*, 5(4):305–313, 2016.
3. Abhijit Das and *et al.*,. Thai automatic signature verification system employing textural features. *IET Biometrics*, 7(6):615–627, 2018.
4. M. A Ferrer and *et al.*,. Multiple training-one test methodology for handwritten word-script identification. In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 754–759. IEEE, 2014.
5. Hemmaphan Suwanwiwat and *et al.*,. Benchmarked multi-script thai scene text dataset and its multi-class detection solution. *Multimedia Tools and Applications*, 80(8):11843–11863, 2021.
6. booktitle=2019 International Conference on Document Analysis and Recognition (ICDAR) pages=987–992 year=2019 organization=IEEE Keserwani, P et al. Zero shot learning based script identification in the wild.
7. Debashis Ghosh and *et al.*,. Script recognition—a review. *IEEE Transactions on pattern analysis and machine intelligence*, 32(12):2142–2161, 2010.
8. A et al. Bhunia. Indic handwritten script identification using offline-online multimodal deep network. *Information Fusion*, 57:1–14, 2020.
9. Kurban Ubul and *et al.*,. Script identification of multi-script documents: a survey. *IEEE Access*, 5:6546–6559, 2017.
10. Sk Md Obaidullah and *et al.*,. Handwritten indic script identification in multi-script document images: A survey. *IJPRAI*, 32(10):1856012, 2018.
11. Sylvie Brunessaux and *et al.*,. The maurdor project: improving automatic processing of digital documents. In *DAS*, pages 349–354. IEEE, 2014.

12. Pawan Kumar Singh and *et al.,*. Benchmark databases of handwritten bangla-roman and devanagari-roman mixed-script document images. *Multimedia Tools and Applications*, 77(7):8441–8473, 2018.
13. Nabin Sharma and *et al.,*. Icdar2015 competition on video script identification (cvsi 2015). In *ICDAR*, pages 1196–1200. IEEE, 2015.
14. booktitle=ICDAR pages=1582–1587 year=2019 organization=IEEE Nayef, Nibal and*et al.,*. Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019.
15. Nibal Nayef and *et al.,*. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In *ICDAR*, volume 1, pages 1454–1459. IEEE, 2017.
16. Deepak Kumar and *et al.,*. Multi-script robust reading competition in icdar 2013. In *4th International Workshop on Multilingual OCR*, pages 1–5, 2013.
17. Chawki Djeddi and *et al.,*. Icdar2015 competition on multi-script writer identification and gender classification using 'quwi'database. In *ICDAR*, pages 1191–1195. IEEE, 2015.
18. Miguel A Ferrer; *et al.,*. MDIW-13 multiscript document database, 2019.
19. Miguel A and*et al.,* Ferrer. Mdiw-13: New database and benchmark for script identification.
20. *al.,* Hu, T. See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification. *arXiv preprint arXiv:1901.09891*, 2019.
21. Ning Zhang and *et al.,*. Part-based r-cnns for fine-grained category detection. In *European conference on computer vision*, pages 834–849. Springer, 2014.
22. *al.,* Wei, X. Mask-cnn: Localizing parts and selecting descriptors for fine-grained image recognition. *arXiv preprint arXiv:1605.06878*, 2016.
23. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
24. Kaiming He and *et al.,*. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
25. *al.,* Long, J. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.
26. *al.,* Chen, L. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018.
27. Hang and*et al.,* Zhang. ResNeSt: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020.
28. *al.,* Liu, Z. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.
29. *al.,* Sun, K. Deep high-resolution representation learning for human pose estimation. In *CVPR*, pages 5693–5703, 2019.
30. *al.,* Yuan, Y. Object-contextual representations for semantic segmentation. *arXiv preprint arXiv:1909.11065*, 2019.
31. *al.,* Dai, J. Deformable convolutional networks. In *CVPR*, pages 764–773, 2017.
32. *al.,* Lee, D . Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICMLW*, volume 3, 2013.
33. *al.,* Lin, T. Focal loss for dense object detection. In *CVPR*, pages 2980–2988, 2017.
34. Canjie and*et al.,* Luo. Learn to augment: Joint data augmentation and network optimization for text recognition. In *CVPR*, pages 13746–13755, 2020.
35. G. Huang and *et al.,*. Densely connected convolutional networks. In *CVPR*, pages 2261–2269, 2017.

36. Sanghyun Woo and *et al.,*. Cbam: Convolutional block attention module. In *ECCV*, pages 3–19, 2018.
37. et *al.,* He, K. Deep residual learning for image recognition. pages 2980–2988, 2017.
38. Quoc Le. Mingxing Tan. Efficientnet: Rethinking model scaling for convolutional neural networks. pages 6105–6114, 2019.
39. Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946, 2019.
40. Phillip Isola and *et al.,*. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.