Deterministic Routing between Layout Abstractions for Multi-Scale classification of Visually Rich Documents

Ritesh Sarkhel[*] and **Arnab Nandi**

Department of Computer Science & Engineering, The Ohio State University

{sarkhel.5, nandi.9}@osu.edu

Abstract

Classifying heterogeneous visually rich documents is a challenging task. Difficulty of this task increases even more if the maximum allowed inference turnaround time is constrained by a threshold. The increased overhead in inference cost, compared to the limited gain in classification capabilities make current multi-scale approaches infeasible in such scenarios. There are two major contributions of this work. First, we propose a *spatial pyramid model* to extract highly discriminative multi-scale feature descriptors from a visually rich document by leveraging the inherent hierarchy of its layout. Second, we propose a *deterministic routing scheme* for accelerating end-to-end inference by utilizing the spatial pyramid model. A depth-wise separable multi-column convolutional network is developed to enable our method. We evaluated the proposed approach on four publicly available, benchmark datasets of visually rich documents. Results suggest that our proposed approach demonstrates robust performance compared to the state-of-the-art methods in both classification accuracy and total inference turnaround.

1 Introduction

Identifying similar documents from a collection of heterogeneous visually rich documents has garnered the intrigue of researchers for a long time. Along with traditional linguistic cues, these documents also use a number of visual modifiers [Sarkhel and Nandi, 2019] to augment/highlight the semantics of different visual areas appearing in them. Whether searching in a digitized library of historical manuscripts, extracting structured records from official ledgers to create a knowledge-base or interacting with a restaurant-menu in an augmented-reality setting, identifying a document as one of N predefined categories is an important precursor to a number of tasks, including indexing, recommendation, transcription, and information extraction.

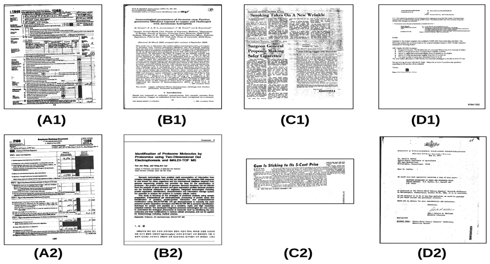
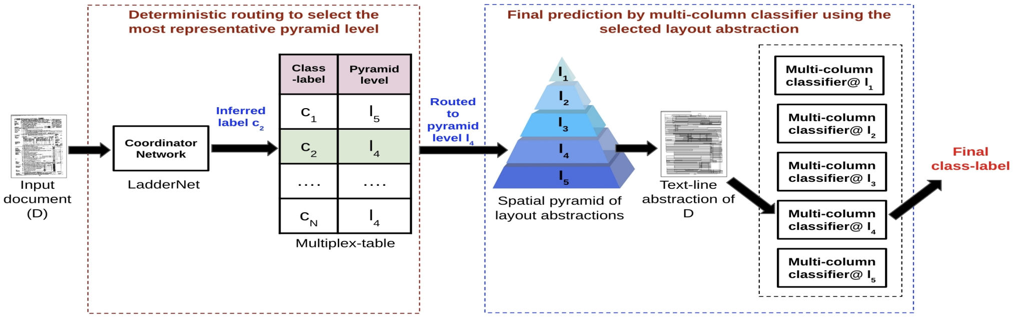


Figure 1: Sample images of visually rich documents from our datasets, demonstrating some of the major difficulties in classifying such documents accurately. Documents A1, A2 from the NIST special dataset-6 and B1, B2 from the MARG dataset belong to different document classes although their layout similarity is high. On the other hand, documents C1, C2 from the Tobacco Litigation dataset and D1, D2 from the RVL-CDIP dataset exhibit low layout similarity despite belonging to the same class

Challenges. To classify visually rich documents, we need to encode local invariant patterns appearing in documents belonging to each class. If the documents are heterogeneous i.e., the content, layout or format of the test document is not known beforehand, this becomes a challenging task, as documents may exhibit high intra-class and/or low inter-class variance (refer to Fig. 1) in layout similarity. If the content of the document is not known, relying on its semantic properties for this purpose may not be feasible. Generating high-quality transcription of real-world documents is also not a trivial task [He and Schomaker, 2017; Sarkhel *et al.*, 2016; 2017]. We explore a layout-based approach for classifying heterogeneous visually rich documents in this work. It has been established in recent past [Gordo *et al.*, 2013; Lazebnik *et al.*, 2006; Lowe, 2004; Mao *et al.*, 2018] that scale-invariant local patterns encoded by multi-scale feature descriptors outperform its single-scale counterparts [Gordo *et al.*, 2013; Bagdanov and Worring, 2001; Afzal *et al.*, 2015] in this scenario. Identifying local invariant patterns persisting at multiple resolutions of a document offers the flexibility to search for pairwise, structurally consistent matches[1] between components across various positions of a document. Few

[*] Contact Author

[1] the one-to-one mapping between structural components of two documents such that the parallel connectivity [Forbus *et al.*, 1995] is preserved [Kumar *et al.*, 2014]


Deterministic routing to select the most representative pyramid level
Input document (D)
Coordinator Network
LadderNet
Inferred label c_2
Routed to pyramid level l_4
Final prediction by multi-column classifier using the selected layout abstraction
l_1
l_2
l_3
l_4
l_5
Spatial pyramid of layout abstractions
Text-line abstraction of D
Multi-column classifier@ l_1
Multi-column classifier@ l_2
Multi-column classifier@ l_3
Multi-column classifier@ l_4
Multi-column classifier@ l_5
Final class-label

Class -label	Pyramid level
c_1	l_5
c_2	l_4
....	....
c_N	l_4

Multiplex-table

Figure 2: End-to-end workflow of our classification scheme. Upon input, the squared rendered image of a test document (D) is fed to the coordinator network. It computes the prediction probabilities of this document image over all class-labels defined for the corpus. The top-1 predicted label (c_2) is used to query a Multiplex-table that returns the optimal layout abstraction level (l_4) to encode D. The document is then converted to a binary image ($I_{l_4,D}$) annotating structural components at l_4 (text-line) layout abstraction. A multi-column classifier trained on text-line abstractions of the training corpus finally extracts feature descriptors from $I_{(l_4,D)}$ and infers the final class-label for D

works have explored this approach to classify visually rich documents by extracting multi-scale feature descriptors in recent past. The first work to utilize this approach successfully was Cesarini et al. [Cesarini *et al.*, 2001]. They proposed a recursive X-Y tree based encoding technique to generate fixed-length signatures, representing each document. Later, Kumar et al. [Kumar *et al.*, 2014] proposed a static zoning scheme, recursively partitioning each document using vertical and horizontal grids and aggregating features extracted from each zone to encode a document. Although not for classification, Xu et al. [Xu *et al.*, 2018] took a similar approach for layout analysis of historical documents in a multi-task setting using an ensemble of fully-connected convolutional neural networks. More recently, Das et al. [Das *et al.*, 2018] have shown that an ensemble of region-based classifiers can also be employed for this task. However, there are a number of limitations in using the existing multi-scale feature descriptors for modern workflows.

One of the major limitations of employing contemporary multi-scale classifiers for real-world applications is that the gain in representational capabilities using multi-scale features is marginal compared to the overhead in end-to-end inference turnaround, making it impractical to deploy them in scenarios where the average inference time is constrained (e.g. augmented reality applications, gesture-based interactive workflows) by a threshold (≈500ms) [Jiang *et al.*, 2018]. Hence, for multi-scale classifiers to be considered feasible solutions in such applications, they need to satisfy two conditions:

C1.1: Representational capability of the feature descriptors should be high

C1.2: End-to-end inference turnaround should be low

Our hypotheses. We hypothesize that the main reason behind the limited representational capability of current aggregation-based approaches to compute multi-scale feature descriptors is large semantic gaps between feature descriptors computed at different resolutions. As most of the feature descriptors computed at a lower resolution are not reused at higher abstractions, we miss out on the opportunity to persist scale-invariant local patterns in the layout, commonly observed in a visually rich document. To address this (condition C1.1), we propose a warm-start approach for computing the multi-scale features in this work. We developed a spatial pyramid model leveraging the inherent layout hierarchy of a document for this purpose. We discuss the spatial pyramid model in more details in Section 2. Our key insight here is that if the layout hierarchy between structural components at various levels of layout abstraction can be preserved, we can potentially recycle features extracted from the lower resolutions at higher levels of abstraction, leading to scale-invariant, highly discriminative feature descriptors to encode the document. We also hypothesize that compared to current aggregation based approaches, end-to-end inference turnaround (condition C1.2) can be minimized if feature descriptors are extracted from the most discriminative level, leveraging a level-wise competition within the spatial pyramid model. We discuss this in more details in Section 3.

Technical contributions. Our first contribution is a spatial pyramid model to represent a visually rich document by leveraging its inherent layout hierarchy. A supervised coordinator network, called *LadderNet*, is responsible for routing the workflow towards the most discriminative level (say l) to encode the document by introducing a level-wise competition within the pyramid. Each level of the spatial pyramid corresponds to a layout abstraction, defined by our layout model (Section 2.1). Once the feature descriptors corresponding to the selected level have been computed, classification is performed using a depth-wise separable [Howard *et al.*, 2017] multi-column convolutional network. Compared to the current aggregation-based approaches, our routing based approach offers two distinct advantages. First, it helps reduce redundant contributions from feature descriptors extracted at multiple scales by maintaining the information about layout hierarchy at each level of the spatial pyramid. Second, compared to the current aggregation-based approaches, it also helps reduce the turnaround time for end-to-end inference (condition C1.2). An overview of the coordinator network and the proposed deterministic routing strategy used to compute multi-scale feature descriptors is presented in Section 3. The multi-column classifier used for final prediction will be discussed in Section 4. To summarize, the major contributions of our work are as follows:

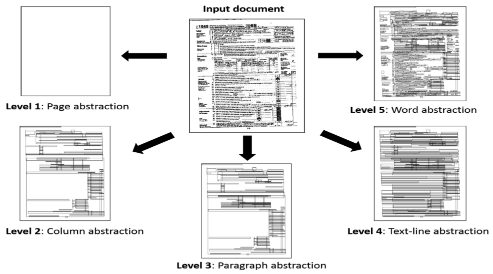

Input document
Level 1: Page abstraction
Level 5: Word abstraction
Level 2: Column abstraction
Level 4: Text-line abstraction
Level 3: Paragraph abstraction

Figure 3: The spatial pyramid model for a sample document from one of our datasets. Each level of the spatial pyramid corresponds to the anonymized equivalent image at its corresponding level of layout abstraction. The anonymized equivalent images at paragraph and column-abstraction of this document are identical because it has a single-column layout. The page-abstraction represents a single bounding-box covering the visual area of the entire document

- A spatial pyramid model to define layout hierarchy at multiple resolutions of a visually rich document
- A level-competitive routing strategy to compute feature descriptors leveraging the spatial pyramid model

Summary of results. We evaluated the proposed classification scheme on four publicly available datasets of single-page visually rich documents. Results suggest that we are able to outperform current state-of-the-art approaches utilizing either handcrafted or deep convolutional network based multi-scale features in terms of both classification accuracy and total inference turnaround time, on all datasets.

2 The Spatial Pyramid Model

We construct a *spatial pyramid model* to facilitate the computation of feature descriptors for each input document.

Definition 2.1. The spatial pyramid model of a visually rich document is a hierarchical construct of structural components at various layout abstractions of the document. Each level of the pyramid corresponds to an *anonymized equivalent image* at that level of abstraction. The layout model used to define these abstractions is discussed in the following section.

2.1 The Document Layout Model

We represent a visually rich document D as a nested tuple (A_D, T_D), where $A_D = \bigcup_i a_i$ denotes the set of *atomic elements* (a_i) appearing in the document and T_D represents the visual organization of D. Each 'word' in D denotes an atomic element in our layout model. We represent the visual organization of a document using a tree-like structure. The leaf-nodes of T_D represent atomic elements of D, whereas its root-node represents the entire document. The non-leaf nodes represent structural components of D at various levels of layout hierarchy. Each node (n_i) in the layout-tree is represented as a nested tuple, $n_i = (x_i, y_i, h_i, w_i, t_i)$; x_i, y_i, h_i and w_i denote the coordinates of the top-left corner, the height and the width of the smallest bounding-box enclosing the visual area and t_i represents the textual content of the node. An edge between a parent and its child node in T_D signifies that the visual area represented by the child node is enclosed by the visual area represented by the parent node. Therefore, the non-leaf nodes in T_D are nested. We define five levels of layout hierarchy using a standard specification format (hOCR [Breuel, 2007]). In this format, every *page* of a document is split into *columns*, every *column* is split into *paragraphs*, every *paragraph* is divided into *text-lines* and every *text-line* is divided into *words*. The layout-tree T_D is defined leveraging this hierarchical relationship. We construct T_D recursively, using a popular open-source page segmentation algorithm [Smith, 2007]. The layout model of a document is constructed offline, before introducing it to our workflow.

Input	Operator	n	c	m	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	inv-bottleneck	1	16	1	1
$112^2 \times 16$	inv-bottleneck	6	24	2	2
$56^2 \times 24$	inv-bottleneck	6	32	3	2
$28^2 \times 32$	inv-bottleneck	6	64	4	2
$14^2 \times 64$	inv-bottleneck	6	96	3	1
$14^2 \times 96$	inv-bottleneck	6	160	3	2
$7^2 \times 160$	inv-bottleneck	6	320	1	1
$7^2 \times 320$	1×1 conv2d	-	1280	1	1
$7^2 \times 1280$	7×7 avgpool	-	-	1	-
$1^2 \times 1280$	1×1 conv2d	-	c'	-	

Table 1: An overview of the LadderNet architecture; Each row denotes a sequence of one or more identical layers, repeated 'm' times. Layers in the same sequence have the output channel size 'c'. The first layer of each sequence has stride 's' and all others use a stride of 1. 'n' is the expansion factor applied to the input dimensions of each inverted bottleneck layer. c' denotes the number of output channels. The architecture of an inverted bottleneck layer is shown in Table 2. 3×3 kernels are used for all spatial convolutions

Input	Operator	Output
$h \times w \times c$	1×1 conv2d + ReLU	$h \times w \times nc$
$h \times w \times nc$	3×3 dw_s + ReLU	$\frac{h}{s} \times \frac{w}{s} \times nc$
$\frac{h}{s} \times \frac{w}{s} \times nc$	1×1 pointwise conv2d	$\frac{h}{s} \times \frac{w}{s} \times c'$

Table 2: Architecture of the depth-separable convolutional block with inverted residual connections, input channel size = c, output channel size = c', stride size = $s \times s$, and expansion factor = n, as mentioned in Table 1; We have used a constant value of $n = 6$ for all layers except the input layer in our implementation

2.2 Constructing the Spatial Pyramid Model

Based on our definitions above, the spatial pyramid model of a visually rich document represents a hierarchical structure with five levels (at most), where each level corresponds to an anonymized equivalent image at a specific level of layout abstraction. The bottom-most level of this structure corresponds to word-abstraction and the top-most level represents the page-abstraction of a document. An illustration of each level of the spatial pyramid for a sample document from one of our datasets is shown in Fig. 3. Once the spatial pyramid is constructed, a coordinator network, called LadderNet, selects the most discriminative level in the spatial pyramid to compute feature descriptors for encoding the input document. We will discuss the routing scheme in details in the following section.

3 Deterministic Routing to a Pyramid Level

One of the major contributions of this work for minimizing the end-to-end inference turnaround is a deterministic routing strategy to select a level in the spatial pyramid model that offers the maximum discriminative capability. The feature descriptors used to encode a test document for final prediction is extracted from the anonymized equivalent image corresponding to the selected level by a multi-column classifier. The key enabler in this is a *Multiplex-table* (refer to Fig. 2).

Definition 3.1. For a corpus with N classes, the Multiplex-table $M = \bigcup_{i,j}(c_i, l_j), \forall i, j$, is a hash-table of dimension $N \times 2$, indexed on class-labels. The first column (key) of each row represents a class-label c_i, $1 \leq \text{i} \leq N$, whereas the second column (value) denotes the optimal pyramid level l_j, $1 \leq \text{i} \leq 5$ corresponding to that class-label.

3.1 Populating the Multiplex-table

Populating the Multiplex-table is the responsibility of a coordinator network. This is a softmax classifier, trained on anonymized equivalent images of the training corpus. The table is populated by introducing level-wise competition within the spatial pyramid. The coordinator network is trained on anonymized equivalent images corresponding to each level of the pyramid, separately, for this purpose. The five-fold cross-validation accuracy for each class-label c_i, $1 \leq \text{i} \leq N$ is computed for each level of abstraction. The pyramid level corresponding to the abstraction that performs the best for a class-label $c_i, \forall i$ is selected as the destination level for c_i in the table. Population of the Multiplex-table for a training corpus is done offline, as a preprocessing step. It is worth mentioning here that we performed exploratory experiments with a number of different variants of the Multiplex-table. The best performing version was empirically selected based on (a) the best cross-validation accuracy and (b) average inference turnaround time achieved on all datasets. We also performed an ablation study in Section 5.3 to investigate the contributions of the routing scheme on end-to-end performance.

3.2 Querying the Multiplex-table

When a test document is presented to our system, a squared (224 × 224), rendered image of the document is fed to the coordinator network. This is a deep convolutional network, trained on rendered images of documents in the original training corpus. To select the optimal pyramid level, the top-1 predicted label (c) by the coordinator network is used to look up the Multiplex-table. The corresponding pyramid level, $l = M(c)$ is the optimal pyramid level for the document. The anonymized equivalent image corresponding to that level in the spatial pyramid model (refer to Fig. 3) is fed to a multi-column classifier to encode and infer the final class-label of the document. *LadderNet*, the coordinator network developed for this purpose is discussed in the following section.

3.3 The Coordinator Network

We developed LadderNet, a supervised coordinator network for selecting the most discriminative layout abstraction to represent a document for the final classification task. An overview of its architecture is presented in Table 1.

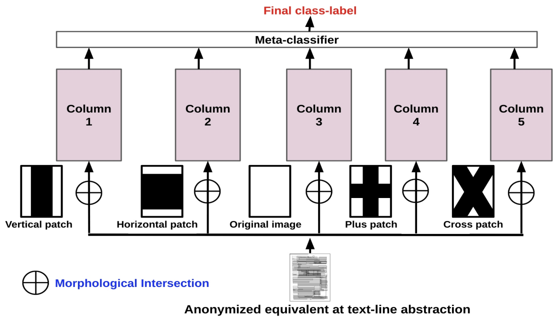


Figure 4: A block-diagram of our proposed multi-column classifier. Each column is a depth-separable convolutional network with inverted residual connections that takes a unique morphologically transformed version of the anonymized equivalent of a test document as input; The image filters used for this are shown in-place

Architecture. The input to LadderNet is a squared, rendered image (224 × 224 × 3) of a visually rich document. It outputs a distribution of prediction probabilities over N class-labels defined for the corpus. As minimizing end-to-end inference turnaround is one of the main objectives (refer to C1.2, Section 1) of this work, we utilize depth-separable convolutional blocks [Howard *et al.*, 2017; Sandler *et al.*, 2018] with inverted residual connections (i.e., skip-connections joining two bottleneck layers) to construct a fast and memory-efficient network for this purpose. Each convolutional block is followed by batch-normalization [LeCun *et al.*, 2015] and a Rectified Linear Unit [LeCun *et al.*, 2015]. The architecture of each convolutional block is shown in Table 2. A logistic regression based classifier is used to compute the prediction probabilities. For each input document, the top-1 prediction is then used to query the Multiplex-table. The anonymized equivalent image at the selected level of layout abstraction is fed to the multi-column classifier for inferring the final class-label for the input document.

Training. LadderNet is trained on rendered images of the visually rich documents in the original training corpus. It is initialized with pre-trained ImageNet weights (L1-transfer of weights [Pan and Yang, 2010]). To address the issue of imbalanced class distribution in the training corpus, we trained LadderNet on a focal-loss [Lin *et al.*, 2018] based learning objective. If p_i denotes the predicted probability of a test document belonging to class-label c_i and y_i represents the corresponding indicator variable (1 if the document belongs to class c_i, 0 otherwise), the loss function is defined as follows:

$$FL(p, y) = -\Sigma_{i=1}^{N} y_i \times (1 - p_i)^{\gamma} \times ln(p_i) \qquad (1)$$

Parameters. In Eq. 1, N is the number of distinct class-labels defined for the corpus, the value of γ is set to 4 for all of our experiments. We closely followed the parameter settings of the MobileNetV2 architecture [Sandler *et al.*, 2018] in our implementation. RMSProp [LeCun *et al.*, 2015] is used for parameter training. The learning parameters *momentum*, *initial-learning-rate*, and *learning-rate-decay* were set to 0.9, 0.045 and 0.98/epoch. This network has the computational cost of approximately 300M multiply-adds.

4 Multi-column Classifier for Final Prediction

Once the optimal pyramid level has been selected (say l), predicting the final class-label for a test document is the responsibility of a multi-column classifier, trained on anonymized equivalent images of the training corpus at layout abstraction corresponding to the selected pyramid level l (refer to Fig. 2). We developed a depth-separable, multi-column convolutional architecture for this purpose. There are five columns in our architecture. Each column is a depth-separable convolutional network with inverted residual blocks, identical in architecture to the coordinator network, introduced in Section 3.3. The input to the multi-column classifier is the anonymized equivalent image of the test document at layout abstraction corresponding to the l^{th} level. Each of its five columns extracts unique feature descriptors from its input and computes a distribution of prediction probabilities over N class-labels defined for the corpus. These individual predictions are then combined using a meta-classifier to output the final class-label inferred for that document.

4.1 Training the Multi-column Classifier

Each of the five columns in our multi-column architecture is trained in parallel, on separate, morphologically intersected [Patin, 2003] versions of the training corpus at l^{th} level of layout abstraction. The image filters (refer to Fig. 4) used to generate these morphologically transformed images for each column have dimensions 224×224. We empirically selected the width of the 'black-bands' in the vertical, plus & cross shaped filters as $w_1 = 0.4 \times \frac{h+w}{2}$ and $w_2 = 0.6 \times \frac{h+w}{2}$ for the horizontal patch filter, where h, w = 224. The training corpus for each column is constructed by morphologically transforming the anonymized equivalent images of documents in the original training corpus, at the l^{th} layout abstraction. Learning objective, weight initialization and parameter values to train each column are similar to what has been described in Section 3.3. During inference, the anonymized equivalent image of a test document and its morphological transformations are fed to appropriate columns (Fig. 4) in the multi-column architecture. Prediction probabilities computed by each column are then aggregated by a *meta-classifier* to output the final class-label.

4.2 Column Aggregation

Let, P_i^j denotes the N-dimensional vector of prediction probabilities computed for the i^{th} document by the j^{th} column in our architecture. Here, N denotes the number of class-labels defined for the corpus. The objective of the meta-classifier is to learn a mapping $\boldsymbol{f} : \Re^{N\times 5} \rightarrow \Re$. The domain of $\boldsymbol{f}$, say Σ, is the aggregate space of prediction probabilities (say $\beta_i, i = 1$ to 5) by each column in our architecture i.e. $\Sigma = \bigcup_{i=1}^{5} \beta_i$. We have investigated a number of different methods in our experimental setup to learn f. This includes simple and weighted averaging, logistic regression and a three-layer (256-256-N) multi-layer-perceptron.

Dataset	Size	No. of classes	σ_{inter}	σ_{intra}
NIST	5595	20	H	L
MARG	1553	9	M	M
Tobacco	3482	10	L	H
RVL-CDIP	400,000	16	L	H

Table 3: An overview of the datasets and their properties used in this work; 'σ_{inter}' and 'σ_{intra}' refer to inter-class and intra-class variance in layout similarity between documents in the dataset; 'H', 'M', 'L' represent High, Medium and Low respectively

5 Experiments

We seek to answer three key questions in this study, (a) how does our method compare against current state-of-the-art methods in terms of classification accuracy? (b) how does it compare in terms of average turnaround time for end-to-end inference? (c) what are the contributions of individual components in our proposed workflow on downstream classification accuracy? To answer the first two questions, we compare our method (Section 5.2) against four state-of-the-art multi-scale classifiers in Section 5.2. To answer the third question, we performed an ablation study in Section 5.3.

5.1 Experiment Design

We evaluated our proposed method on four publicly available benchmark datasets of single-page visually rich documents. These are: the NIST special dataset-6 (**D1**) [NIST, 2018] (filled tax-forms from the IRS-1040 package, 1988), the Medical Article Records Groundtruth (MARG) dataset (**D2**) [Thoma, 2003] (front pages of scanned biomedical journals from the National Library of Medicine), the RVL-CDIP dataset (**D3**) [Harley *et al.*, 2015] (scanned images from the IIT-CDIP [Lewis *et al.*, 2006] collection, containing document categories such as 'letter', 'memo', 'email', 'form', 'invoice' etc.) and the Tobacco litigation dataset (**D4**) [Kumar *et al.*, 2014] (a sample of 3482 documents from the IIT-CDIP collection). A brief overview of some of the important properties of these datasets is shown in Table 3. Sample images from each dataset are shown in Fig. 1. To ensure fair comparison, we follow similar experimental designs as suggested by previous researchers [Kumar *et al.*, 2014; Das *et al.*, 2018]. For the NIST special dataset-6, we construct the training corpus by randomly selecting two documents from each class. The rest of the documents were used to construct the test corpus. For the Tobacco litigation dataset, $10 < n <$100 documents were selected to construct the training corpus from each class while the rest of the documents comprised the test corpus. 20% of documents were randomly selected from each document class to construct the training corpus for the MARG dataset. For the RVL-CDIP, the entire dataset was partitioned in 80:20 ratio to construct the training and test corpus. For every dataset, experiments were repeated 25 times, each time with a randomly selected partition of the dataset. All experiments were performed on a 12GB NVIDIA Titan-XP GPU. The best model based on cross-validation was used to report the test accuracy at each trial.

We measure the performance of our proposed methodology using two evaluation metrics. First, we measure the median classification accuracy obtained for all datasets

Index	NIST (D1)	MARG (D2)	Tobacco (D3)	RVL-CDIP (D4)
A1	94.30	70.10	41.85	48.72
A2	100.0	75.25	43.10	50.25
A3	100.0	**95.05**	60.58	71.02
A4	100.0	82.60	78.25	92.21
B_{avg}	100.0	83.95	80.92	89.86
B_{wavg}	100.0	84.05	81.25	90.43
B_{log}	100.0	84.81	82.50	90.18
B_{mlp}	**100.0**	85.25	**82.78**	**92.77**

Table 4: A comparative analysis of classification accuracy obtained for all datasets against state-of-the-art methods

Index	Speedup in inference turnaround (δ)
A1	0.84
A2	1.58
A3	2.07
A4	6.83

Table 5: Speedup achieved by our proposed method over state-of-the-art methods in total inference turnaround

over 25 trials. Second, we measure the relative speedup in inference turnaround time using our proposed method compared to a baseline. The speedup-factor ($\boldsymbol{\delta}$) computed for this purpose is defined as follows, $\boldsymbol{\delta} = \frac{t_1}{t_2}$, where t_1 and t_2 denote the inference turnaround for a baseline method and our proposed method respectively, averaged over all datasets. The median accuracy of our classification scheme, computed using four different meta-classifiers i.e., simple averaging (B_{avg}), weighted averaging (B_{wavg}), logistic regression (B_{log}) and multi-layer perceptron (B_{mlp}) is reported in Table 4. Our entire workflow consists of approximately 600 million multiply-adds. The average turnaround time which include: (a) selecting the optimal pyramid level for a test document by feeding it to the coordinator network, (b) generating the anonymized equivalent image and (c) inferring the final class-label is approximately 362 ms ($\pm$10.27 ms), well within the threshold ($\approx$500 ms) set by a number of modern interactive workflows.

5.2 Comparison Against Existing Methods

We compared the end-to-end performance of our method against four contemporary multi-scale approaches. These baseline methods were selected from the existing literature, if (a) it reports state-of-the-art result on one of our datasets, (b) differs significantly from the previous methods. Input to each method is the squared rendered image of a document. Same experimental protocols, described in Section 5.1, were followed for each baseline. The median accuracy and average speedup-factor obtained from these experiments are presented in Table 3 and 4 respectively. For fair comparison, we did not consider any preprocessing steps in the baseline methods when computing their inference turnaround time. Our first competitor (A1) [Cesarini *et al.*, 2001] proposes a modified XY-tree based encoding technique to represent each document using carefully designed handcrafted features. We outperformed this method significantly in terms of classification accuracy on all datasets. Our second baseline (A2) [Kumar *et al.*, 2014] utilizes vertical and horizontal pooling operations to compute multi-scale SURF descriptors [Bay *et al.*, 2006] for encoding each document. Significant improvement was observed over this baseline for datasets D2, D3 and D4. Our third competitor (A3) [Gordo *et al.*, 2013] proposed multi-resolution run-length histogram features for classifying visually rich documents. Although this approach outperformed our method on dataset D2, its success could not be replicated on other datasets. In fact, we were able to outperform this method on two of the most challenging datasets, D3 and D4. The final baseline method (A4) [Das *et al.*, 2018] in our experimental setup is an ensemble of region-specific CNN's, combining individual predictions to infer the final class-label. We observed significant improvement over this baseline.

Index	S1	S2	S3	Δ**Acc.**(%)			
				D1	D2	D3	D4
S1	X	✓	✓	0.81	2.70	4.74	3.05
S2	✓	X	✓	0.0	1.15	2.80	2.27
S3	✓	✓	X	0.50	1.59	2.35	1.83

Table 6: An overview of results from our ablation studies

5.3 Ablation Study

To investigate the relative contributions of individual components in our workflow on the downstream classification accuracy, we performed an ablation study. Each row in Table 6 measures the effect of a component of our workflow on the downstream classification accuracy, following the same experimental setup in Section 5.1. The final column of this table quantifies the absolute effect of that component on the median accuracy obtained for a dataset. In S1, we measure the contribution of the depth-separable convolutional architecture (as coordinator network and the multi-column classifier) on the downstream classification accuracy. Replacing this with a traditional convolutional [LeCun *et al.*, 2015] architecture results in a significant decrease in accuracy for D2, D3, and D4. To evaluate the contribution of the level-competitive routing strategy, in S2, we concatenated feature descriptors from all levels and trained the multi-column classifier. Improvements were observed over this baseline on datasets D2, D3, and D4. More importantly, an average speedup of 3.85 was observed. In S3, we evaluated the contribution of the multi-column architecture by training a single-column of our architecture on the augmented training corpus. Improvements in performance were significant over this baseline for all datasets.

6 Conclusion

We have proposed a deterministic routing strategy for multi-scale classification of visually rich documents in this work. Comparisons against the existing approaches show that we are able to perform competitively or better on all datasets, achieving significant speedup in inference turnaround time. In the future, we would like to extend this work to an end-to-end trainable, attention-based network to pool information across all levels of the spatial pyramid and derive highly discriminative hybrid representations of each test document.

References

[Afzal *et al.*, 2015] Muhammad Zeshan Afzal, Samuele Capobianco, Muhammad Imran Malik, Simone Marinai, Thomas M. Breuel, Andreas Dengel, and Marcus Liwicki. Deepdocclassifier: Document classification with deep convolutional neural network. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1111–1115, Aug 2015.

[Bagdanov and Worring, 2001] Andrew D. Bagdanov and Marcel Worring. Fine-grained document genre classification using first order random graphs. In *Proceedings of Sixth International Conference on Document Analysis and Recognition*, pages 79–83, 2001.

[Bay *et al.*, 2006] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[Breuel, 2007] Thomas M. Breuel. The hocr microformat for ocr workflow and results. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 1063–1067, Sept 2007.

[Cesarini *et al.*, 2001] Francesca Cesarini, Marco Lastri, Simone Marinai, and Giovanni Soda. Encoding of modified x-y trees for document classification. In *Proceedings of Sixth International Conference on Document Analysis and Recognition*, pages 1131–1136. IEEE, 2001.

[Das *et al.*, 2018] Arindam Das, Saikat Roy, and Ujjwal Bhattacharya. Document image classification with intra-domain transfer learning and stacked generalization of deep convolutional neural networks. *arXiv preprint arXiv:1801.09321*, 2018.

[Forbus *et al.*, 1995] Kenneth D Forbus, Dedre Gentner, and Keith Law. Mac/fac: A model of similarity-based retrieval. *Cognitive science*, 19(2):141–205, 1995.

[Gordo *et al.*, 2013] Albert Gordo, Florent Perronnin, and Ernest Valveny. Large-scale document image retrieval and classification with runlength histograms and binary embeddings. *Pattern Recognition*, 46(7):1898 – 1905, 2013.

[Harley *et al.*, 2015] Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 991–995. IEEE, 2015.

[He and Schomaker, 2017] Sheng He and Lambert Schomaker. Beyond ocr: Multi-faceted understanding of handwritten document characteristics. *Pattern Recognition*, 63:321–333, 2017.

[Howard *et al.*, 2017] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[Jiang *et al.*, 2018] Lilong Jiang, Protiva Rahman, and Arnab Nandi. Evaluating interactive data systems: Workloads, metrics, and guidelines. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1637–1644. ACM, 2018.

[Kumar *et al.*, 2014] Jayant Kumar, Peng Ye, and David Doermann. Structural similarity for document image classification and retrieval. *Pattern Recognition Letters*, 43:119–126, 2014.

[Lazebnik *et al.*, 2006] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[LeCun *et al.*, 2015] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[Lewis *et al.*, 2006] David Lewis, Gady Agam, Shlomo Argamon, Ophir Frieder, D Grossman, and Jefferson Heard. Building a test collection for complex document information processing. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 665–666. ACM, 2006.

[Lin *et al.*, 2018] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[Lowe, 2004] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[Mao *et al.*, 2018] Chaojie Mao, Yingming Li, Yaqing Zhang, Zhongfei Zhang, and Xi Li. Multi-channel pyramid person matching network for person re-identification. *arXiv preprint arXiv:1803.02558*, 2018.

[NIST, 2018] NIST. Nist special database 6. https://www.nist.gov/srd/nist-special-database-6, 2018. Accessed: 2018-09-30.

[Pan and Yang, 2010] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[Patin, 2003] Frédéric Patin. An introduction to digital image processing. *online]: http://www. programmersheaven. com/articles/patin/ImageProc. pdf*, 2003.

[Sandler *et al.*, 2018] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[Sarkhel and Nandi, 2019] Ritesh Sarkhel and Arnab Nandi. Visual segmentation for information extraction from heterogeneous visually rich documents. In *Proceedings of the 2019 International Conference on Management of Data*, pages 247–262. ACM, 2019.

[Sarkhel *et al.*, 2016] Ritesh Sarkhel, Nibaran Das, Amit K Saha, and Mita Nasipuri. A multi-objective approach towards cost effective isolated handwritten bangla character and digit recognition. *Pattern Recognition*, 58:172–189, 2016.

[Sarkhel *et al.*, 2017] Ritesh Sarkhel, Nibaran Das, Aritra Das, Mahantapas Kundu, and Mita Nasipuri. A multi-scale deep quad tree based feature extraction method for the recognition of isolated handwritten characters of popular indic scripts. *Pattern Recognition*, 71:78–93, 2017.

[Smith, 2007] Ray Smith. An overview of the tesseract ocr engine. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 629–633. IEEE, 2007.

[Thoma, 2003] GFG Thoma. Ground truth data for document image analysis. In *Symposium on document image understanding and technology (SDIUT)*, pages 199–205, 2003.

[Xu *et al.*, 2018] Yue Xu, Fei Yin, Zhaoxiang Zhang, and Cheng-Lin Liu. Multi-task layout analysis for historical handwritten documents using fully convolutional networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 1057–1063. AAAI Press, 2018.