Deep Semantic Segmentation of News Articles in JPEG Compressed Domain

Bulla Rajesh[1], Mohammed Javed[1], Anand Kumar[2]
[1]Department of IT, Indian Institute of Information Technology, Allahabad, 211015, U.P, India
[2]Department of EEE, National Institute of Technology, Trichy, India
Emails: rajesh091106@gmail.com; javed@iiita.ac.in; anandknitt@gmail.com;

***Abstract*—The problem of semantic labelling in news articles is still a very challenging research issue in the field of computer vision, because of varying news layouts, diverse styles and shapes of news segments, different languages and fonts, dynamic embedding of images and advertisements based on the regional flavour, etc. In the current digital scenario, since most of the news articles are archived and retrieved directly in the JPEG compressed form, this research paper proposes to accomplish deep learning based semantic segmentation of news articles directly in the JPEG compressed domain without full decompression. The novelty here is to directly feed the JPEG compressed DCT stream into the deep learning model to accomplish the task of semantic labelling with reduced computational cost. Two deep learning models are proposed here- Comp-HRNet (High-Resolution Network) and Comp-FCN (Fully Convolutional Network) for semantic segmentation in the compressed domain. The models are tested on two datasets - first is open source benchmark dataset of Russian news papers, and the second is manually created called IIITA-ANA (Assorted News Articles) dataset. The experimental results show that the proposed models have achieved a state-of-the-art performance on news papers in JPEG compressed domain.**

***Index Terms*—DCT Coefficients, Semantic Segmentation, news Paper Analysis, JPEG Compressed Domain, Article Boundary Localization.**

I. Introduction

Printed news papers have been a rich source of day-to-day information, and also as a historical record ever since their inception [1], [2]. In addition to the text information, they are supported with photographs and advertisements based on the regional flavour, also with attractive fonts, headers and footers [3]. In the current digital scenario, since more and more news articles are archived in the digital form, it is very much necessary to develop computer based technology to automatically analyze news articles. In this context, semantic labelling provides an elemental understanding of all the important segments present in the news articles. However, the problem of semantic labelling in news articles is very challenging because of varying layouts, diverse font styles and shapes of segments, multilingual texts, dynamic background embeddings of images and advertisements as shown in Fig-1. An efficient semantic labelling model can become a very handy digital tool to facilitate quick access to archived news articles [1], [4]–[6], and provide a versatile way to store, index, control the segments on digital platforms [7]. In the DIA literature, Optical Character Recognition (OCR) techniques have been the most used method [4], [8] to read and digitize the news articles. But, before feeding them into the OCR, the locations of the important news segments needs to be identified and labelled. Therefore developing an efficient semantic segmentation model for labelling the news article segments is very crucial, and the same objective is underscored in this research paper.

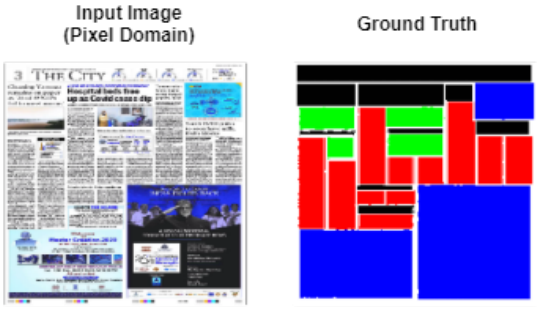


Fig. 1: A sample news article image (on left), with semantic labelling of different news segments (on right) shown with meaningful color codes

Since semantic labelling provides the structural information of the objects present in the image, several studies have been reported on the idea of semantic segmentation in the literature. The early approaches were based on contour extraction [9], texture [10], region [11] and patches [12]. Subsequently, the morphological based operations were reported in [7]. A multi-scale technique using local connectivity was addressed in [13]. A graph-based modeling and structural analysis was discussed in [14] to segment text and non text components. The low level components such as titles, paragraphs, columns, separator lines are identified and clustered using smoothing [15] and connected component analysis [16]. The rule based search for the components is explained in [17]. Most of these handcrafted models fail in segmenting objects in the complex layouts and in the presence of variable illumination [4]. During the present Machine Learning (ML) realm, various segmentation methods have been introduced based on the idea of support vector machines in [18] and neural networks [19]. However, to the best of our knowledge, these techniques perform poorly on the news articles in complex layouts as reported by [4].

Recently, the problem of semantic labelling has been attempted by different deep learning architectures such as U-net [20], SegNet [21], EDANet [22] and HRNet [23], [24] on scenery images, and FCN on historical document images [6], [25]. Most of them are encoder-decoder architectures where the encoder network downscales the feature map as they go deeper to capture the high semantic features set. Then the decoder network uses that downscaled feature set and recovers

the high resolution semantic map of the pixel by upscaling. In the process of building the high resolution prediction map from low resolution feature set, it loses the significant shape and boundary details which are very important to improve the efficiency. In continuation to such models [25] another architecture Mask R-CNN is proposed in [4] to handle the non-rectangular shaped objects and to detect type of objects on news papers. Recently, a high resolution network (HRNet) to maintain the high resolution feature set across the network is proposed in [23]. This model designed the parallel layers to keep and forward the high resolution feature set in various scales to next layers in the network. To further increase the accuracy, this model has been optimized by introducing the mixed dilated convolution module for better recognizing the shapes of the object and multi-level data dependent feature aggregation (MDFA) for identifying the smaller objects with fuzzy shapes [24]. Though this model achieved better performance, it needed the maintenance of many parallel layers with large memory buffer size that required additional computational cost. However, to the best of our knowledge, an optimized deep learning based model for semantic labelling of news articles directly in compressed domain in not yet explored, and hence our objective is to explore novel deep learning models to accomplish semantic segmentation with JPEG compressed documents.

In the present era of Big Data, archiving large volumes of news papers in the raw format will consume a huge space, and hence they are subjected to compression for storage and bandwidth efficiency [26]–[28]. JPEG is the most commonly used digital image format, and is supported as default compressed format in all the present digital and mobile devices, because it provides a good trade off between compression ratio and image quality [29], [30]. Due to this fact, we assume that most of the news papers are available in JPEG compressed format, and hence developing techniques to directly analyze the JPEG compressed images without involving decompression is very much significant.

Working on JPEG compressed representation has been a topic of research interest since early 90's [31], and most of methods developed were handcrafted and proposed on DCT representation for various tasks such as low and high level feature extraction in [31] and in [32], segmentation [33], retrieval [34], and enhancement [35]. In [33], morphological operation were used on Encoding Cost Map (ECM) image extracted from DCT compressed stream. Recently, the robustness of deep learning architectures made the classification [36]–[38], object detection [39], segmentation [40], and recognition [41] tasks possible directly on the JPEG compressed data, where DCT compressed stream from the compressed images were fed to the models. A deep residual learning and approximation of different deep learning operations for compressed stream was reported in [42]. The object detection by feeding the DCT stream using different filter setting, for example 8×8, 4×4, in the models is explained in [39]. Similar to the proposed approach, the task of semantic segmentation of road scene image in DCT representation was explored in [43]. This method had proposed a different technique frequency component rearrangement (FCR), where each DCT block $(8, 8, 1)$ was reshaped to $(1, 1, 64)$, which means, the individual frequency value in each block of an image with size$(M, N, 3)$ was placed in a specific channel in the third dimension with size $(M/8, N/8, 192)$. This representation enabled model to exploit the best features through convolution operations, but required a preprocessing stage and increased depth of layers for learning. Recently text-line and word segmentation in JPEG compressed document images was explained in [40], where a DC_Reduced image extracted from the JPEG compressed DCT coefficients was explored. Because of extracting a low resolutions image from compressed stream better performance in terms of computation and memory was achieved. To the best of our knowledge there has been no attempt to address the problem of semantic segmentation directly in JPEG compressed news articles with deep learning models. Whereas the proposed deep learning models target to eliminate the additional preprocessing stage and propose an optimized learning model for semantic segmentation in JPEG compressed domain.

Overall in this research paper, we propose two deep learning architectures - Comp-HRNet and Comp-FCN for semantic segmentation of news articles directly in JPEG compressed domain. The first proposed model is based on the HRNet [23] architecture where the model is redesigned with reduced number of parallel layers and depth of the network to suit the compressed input of the news article images. This Comp-HRNet model is trained to perform the newspaper semantic segmentation at a faster rate and consumes lesser memory as compared to the other base models. Similarly, inspiring from the model in [20], we have also explored the most popular semantic labeling model encoder decoder based architecture-Comp-FCN for the same segmentation task. Since this model contains few dense layers, training and running the model becomes faster. In order to show the efficacy of models two benchmark datasets were used - open source benchmark dataset of Russiannews papers, and manually created IIITA-ANA (Assorted News Articles) dataset. The experimental results showed a state-of-the-art performance in JPEG compressed domain. The remainder of the paper is divided into four sections. Section II discusses the proposed methodology and model architecture. Sections III report the experimental protocols, experimental results, and comparative analysis. Section IV concludes the work with a brief summary of possible future works.

II. Proposed Methodology

This section provides a brief background about extraction of JPEG input stream from compressed images, the mathematical approximation on DCT representation, discussion on proposed methodology, and the details of the proposed architectures.

A. Extraction of JPEG compressed stream

To understand the process of extracting compressed stream and feeding the same into the deep neural network, a brief explanation to steps of the JPEG compression algorithm is provided here. The detailed explanation of this algorithm is reported in [29]. The JPEG algorithm compresses a typical

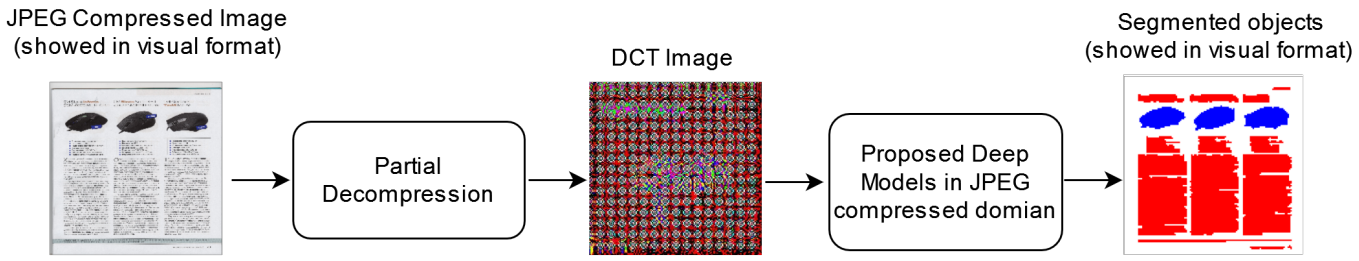


Fig. 2: The extraction of the DCT stream from the JPEG compressed image to feed into the proposed deep learning models for semantic labeling directly in JPEG compressed domain

image with 10:1 compression ratio and maintains a reasonable visual quality. During compression, the raw/uncompressed image R in RGB format with size of $M \times N$ is first transformed into an image I of $YCbCr$ format where Y is luminance and Cb and Cr are chrominance parts using the per-pixel calculation for an i^{th} pixel using Eq.(1):

$$I_i^{\mathsf{T}} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{pmatrix} R_i^{\mathsf{T}} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} \quad (1)$$

Then each channel of image in $YCbCr$ domain is down-sampled and divided into 8×8 blocks and each of them pass through forward Discrete Cosine Transform (DCT) using the equation Eq.(2).

$$D_{rc} = \frac{C_u C_v}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} I(i,j) cos(\frac{(2i+1)u\pi}{16}) cos(\frac{(2j+1)v\pi}{16}) \quad (2)$$

$$Where, \qquad C_u, C_v = \begin{cases} \frac{1}{\sqrt{2}}, & for \;\; u, v = 0 \\ 1, & otherwise, \end{cases}$$

where (D_{rc}) is the DCT transformed block containing DC and AC coefficients and $I(i,j)$ is the input pixel block of size 8×8 from $YCbCr$ channels. Further a sequence of compression steps such as quantization, separate encoding (Run length and followed by entropy) for DC and AC coefficients are applied to convert the DCT transformed image into binary form which is the final form of compressed image. Similarly, during decompression, the same set of inverse operations are applied in reverse order to decompress the compressed contents. Since the JPEG compressed news articles are available in entropy coded format, a partial decompression such as entropy and run length decoding are applied to extract the quantized DCT blocks. This stream is the optimized feature set and sparse in nature, and is fed as input to the proposed deep learning models.

B. Approximation on DCT representation

Since DCT is an orthonormal transform, and achieves best energy compaction and decorrelation of intensities, it has been widely used for encoding image (JPEG, HEIF), digital video (MPEG, H.26x), and digital television (SDTV, HDTV and VOD). Since the operations on the transformed data has wide applications ,therefore, any typical operation on transformed data must be better approximated in DCT domain to build efficient models with increased performance directly in compressed domain. Therefore this subsection attempts to redefine the convolution and sampling operations, based on single 8×8 DCT block, which are the basic operation in the deep learning models, and this DCT block stream is the input to models.

Let $I(M,N)$ be a luminance image, where M and N are the number of rows and columns. If image I is decomposed into a non-overlapping square blocks of size $S \times S$, where $I = \{I_{rc} | r \in [0, M/S-1], c \in [0, N/S-1]\}$, here (r,c) shows the position of the each block I_{rc}. Then the DCT transformation for a block of S pixels can be formulated in matrix form as:

$$D_{rc} = C I_{rc} C^T \quad (3)$$

$$Where, \qquad C(u,i) = \begin{cases} \frac{1}{\sqrt{S}}, & i = 0 \\ \sqrt{\frac{2}{S}} \cos(\frac{(2u+1)i\pi}{2S}) & i \neq 0 \end{cases}$$

Here $N = 8$ and $C(S \times S)$ acts as 8×8 DCT transformation matrix and D_{rc} is the DCT block for a pixel block I_{rc}.

$$\begin{bmatrix} .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 \\ .4904 & .4157 & .2778 & .0975 & -.0975 & -.2778 & -.4157 & -.4904 \\ .4619 & .1913 & -.1913 & -.4619 & -.4619 & -.1913 & .1913 & .4619 \\ .4157 & -.0975 & -.4904 & -.2778 & .2778 & .4904 & .0975 & -.4157 \\ .3536 & -.3536 & -.3536 & .3536 & .3536 & -.3536 & -.3536 & .3536 \\ .2778 & -.4904 & .0975 & .4157 & -.4157 & -.0975 & .4904 & -.2778 \\ .1913 & -.4619 & .4619 & -.1913 & -.1913 & .4619 & -.4619 & .1913 \\ .0975 & -.2778 & .4157 & -.4904 & .4904 & -.4157 & .2778 & -.0975 \end{bmatrix}$$

In Eq.(3) the transformation matrix C is an unitary real valued function, then $C^{-1} = C^T$. Then the inverse DCT transform can be calculated as in Eq.(4):

$$I_{rc} = C^T D_{rc} C \quad (4)$$

Based on the concept of orthogonal transforms theory, the DCT transform matrix C can be seen as the complete set of orthonormal basis vectors of length L, and each of that vector represents a indicated by $c_u^T[L \times 1]$. Where u denote line number in C, that is $c_u = [C(u,0), C(u,1), ..., C(u,L-1)]^T$ and $u = 0, 1, 2, ..., L-1$. Based on this theory the transformation matrix C can be rewritten as:

$$C = \begin{bmatrix} {c_0}^T \\ {c_1}^T \\ \vdots \\ {c_{L-1}}^T \end{bmatrix} \quad (5)$$

Based on defining the transform matrix C in terms of basis vectors c_u above the IDCT transform to get the pixel block I_{rc} for a DCT block D_{rc} is represented as

$$I_{rc} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} D_{rc}(i,j) c_i c_j^T \tag{6}$$

The Eq.(6) explains that the I_{rc} is the weighted sum or linear combination of matrices $c_i c_j^T$, for all $i, j = 0, 1, ..., L-1$. These are the basis image through which the compressed stream has been generated to feed to the proposed models. Representing the basis image $c_i c_j^T$ by $D_{C_{i,j}}[S \times S]$ the Eq.(6) can be rewritten in compact form is:

$$I_{rc} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} D_{rc}(i,j) D_{C_{i,j}} \tag{7}$$

Based on Eq.(4), (5) and (6) the transformation matrix C enables a first type of convolution operation on news articles directly in its compressed representation with close approximation.

1) Approximation: Given a 8×8 pixels in YCbCr domain represented as $R = \{r_{00}, \dots\ r_{77}\}$, let $D = \{d_{00}, \dots\ d_{77}\}$ be the DCT coefficients of R then for $1 \leq m \leq 15$, the approximation of R using the equation 2 given as $\hat{R}$.

The least squared error for the apporximation is given by

$$Error,\ e_m = \sum_{i=0}^{7} \sum_{j=0}^{7} \left(\hat{R}_{ij} - R_{ij} \right)^2 \tag{8}$$

With the help of the above explanation the convolution operation on DCT input block I_{rc} of size $S \times S$ with an odd size kernal K, $K[(2P+1) \times (2Q+1)]$ with P,Q are the positive integers, can be represented as:

$$D(r,c) = I_{r,c} \circledast K = \sum_{p=-P}^{P} \sum_{q=-Q}^{Q} I(r-p, c-q) K(p,q)$$
$$\forall\ u = 0, 1, 2, ..., M-1,\ \ v = 0, 1, 2, ..., N-1 \tag{9}$$

C. Downsampling and Upsampling on DCT representation

Here, we describe the up-scaling and down-scaling of DCT blocks to interpret the behavior of the receptive field of coefficients in different layers of the network and perform scaling similar to RGB images.

1) Downsampling: Let d_1, d_2, d_3, and d_4 represent four consecutive 8×8 blocks and D_1, D_2, D_3, and D_4 represent the DCT values of these blocks. Let $\hat{D}_1$ denote the first 4×4 (low-pass) components from D_1 and so on for $\hat{D}_2$, $\hat{D}_3$, and $\hat{D}_4$. Let $\hat{d}_1$ represent inverse DCT of $\hat{D}_1$ and so on.Then $\hat{d} = \begin{bmatrix} \hat{d}_1 & \hat{d}_2 \\ \hat{d}_3 & \hat{d}_4 \end{bmatrix}$ represents the downsampled low-pass version of $d = \begin{bmatrix} d_1 & d_2 \\ d_3 & d_4 \end{bmatrix}$. Let $\hat{D}$ be the DCT transform of $\hat{d}$. We should now get $\hat{D}$ in terms of $\hat{D}_1$, $\hat{D}_2$, $\hat{D}_3$ and $\hat{D}_4$.

$$\begin{aligned}
\hat{D} &= T \hat{d} T^T \\
&= \begin{bmatrix} T_L & T_R \end{bmatrix} \begin{bmatrix} \hat{d}_1 & \hat{d}_2 \\ \hat{d}_3 & \hat{d}_4 \end{bmatrix} \begin{bmatrix} T_L \\ T_R \end{bmatrix} \\
&= \begin{bmatrix} T_L & T_R \end{bmatrix} \begin{bmatrix} {T_4}^T \hat{D}_1 T_4 & {T_4}^T \hat{D}_2 T_4 \\ {T_4}^T \hat{D}_3 T_4 & {T_4}^T \hat{D}_4 T_4 \end{bmatrix} \begin{bmatrix} T_L \\ T_R \end{bmatrix} \\
&= \left(T_L {T_4}^T\right) \hat{D}_1 \left(T_L {T_4}^T\right)^T + \left(T_L {T_4}^T\right) \hat{D}_2 \left(T_R {T_4}^T\right)^T \\
&+ \left(T_R {T_4}^T\right) \hat{D}_3 \left(T_L {T_4}^T\right)^T + \left(T_R {T_4}^T\right) \hat{D}_4 \left(T_R {T_4}^T\right)^T
\end{aligned} \tag{10}$$

Here T_L and T_R are the first and last four columns of DCT transformation matrix T respectively and T_4 is the DCT transformation matrix for 4×4 block. T_4 is represented as the following 4×4 matrix.

$$\begin{bmatrix} .5 & .5 & .5 & .5 \\ .6532 & .2705 & -.2705 & .6532 \\ .5 & -.5 & -.5 & .5 \\ .2705 & -.6532 & .6532 & -.2705 \end{bmatrix}$$

We see that for $k = 0, 1, 2, 3$ the $2k^{th}$ row of T_L is $\sqrt{2}$ times the k^{th} row of T_4 and the $2k^{th}$ row of T_R is $-\sqrt{2}$ times the k^{th} row of T_4. Since k^{th} rows of T_4 are orthogonal so the $2k^{th}$ rows of T_L and T_R are orthogonal as well. Therefore the matrices $T_L {T_4}^T$ and $T_R {T_4}^T$, have zeros in every $2k^{th}$ row with a expect the k^{th} column of that row. Odd rows of T are anti-symmetric and even rows are symmetric and the vice versa applies for T_4. Using this, we can say that all the values of $T_L {T_4}^T$ and $T_R {T_4}^T$ are identical in magnitude and only vary in sign. Thus $T_L {T_4}^T(i,j) = (-1)^{i+j} T_R {T_4}^T(i,j)$ for $i, j = [0, 7]$.Now, let A be the matrix with terms where $i + j$ is even and B be the matrix with rest of the terms. We have $T_L {T_4}^T = A + B$ and $T_R {T_4}^T = A - B$. Hence we can show that

$$\hat{D} = (X + Y) A^T + (X - Y) B^T \tag{11}$$

where

$$X = A\left(\hat{D}_1 + \hat{D}_3\right) + B\left(\hat{D}_1 - \hat{D}_3\right) \tag{12}$$

$$Y = A\left(\hat{D}_2 + \hat{D}_4\right) + B\left(\hat{D}_2 - \hat{D}_4\right) \tag{13}$$

2) Upsampling: Now for upsampling the DCT image, we can get back $\hat{D}_1$, $\hat{D}_2$, $\hat{D}_3$, and $\hat{D}_4$ from $\hat{D}$. Since the matrices T and T_4 are easily invertible, the following equations will give the upsampled image.

$$\hat{D}_1 = \left(T_L {T_4}^T\right)^T \hat{D} \left(T_L {T_4}^T\right)^T \tag{14}$$

$$\hat{D}_2 = \left(T_L {T_4}^T\right)^T \hat{D} \left(T_R {T_4}^T\right)^T \tag{15}$$

$$\hat{D}_3 = \left(T_R {T_4}^T\right)^T \hat{D} \left(T_L {T_4}^T\right)^T \tag{16}$$

$$\hat{D}_4 = \left(T_R {T_4}^T\right)^T \hat{D} \left(T_R {T_4}^T\right)^T \tag{17}$$

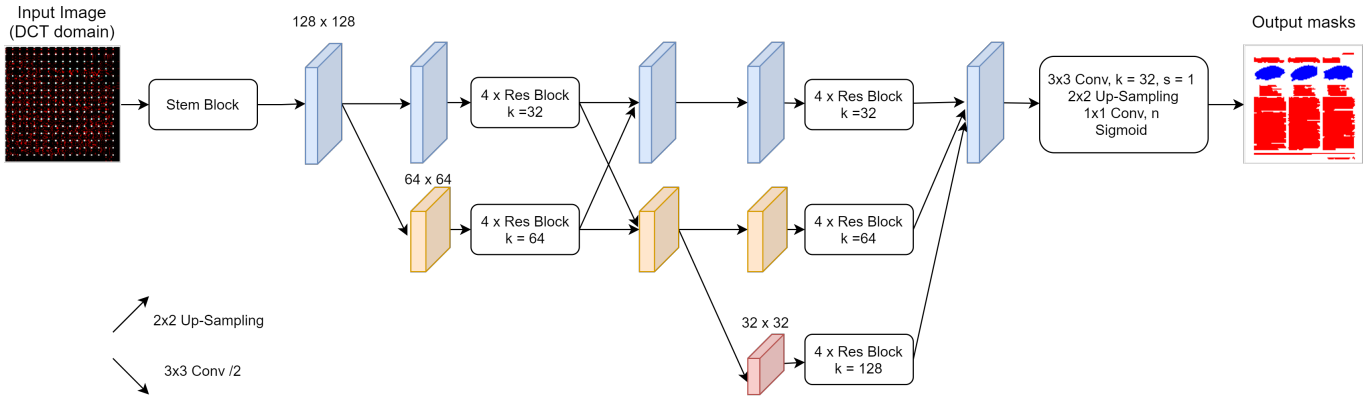


Fig. 3: The proposed Comp-HRNet architecture for semantic segmentation of news articles in JPEG compressed domain.

D. Methodology and Proposed Architectures

The sequence of steps involved in the proposed methodology are summarized by the block diagram shown in Fig-2. The procedure is divided into 2 steps: Firstly a partial decompression is applied on JPEG compressed news paper to obtain the DCT stream, and secondly training the two proposed architectures Comp-FCN and Comp-HRNet by directly feeding DCT coefficients of news papers to perform the segmentation task in JPEG compressed domain.

The proposed models Comp-HRNet and Comp-FCN and their architectures are shown in Fig-3 and Fig-5. The Comp-HRNet model is designed based on using the knowledge of both HRNet model [23] and optimized HRNet model [24]. This model is constructed using minimum number of layers for reducing the computational and memory charges. Similarly, Since FCN is also popular architecture for semantic segmentation, the Comp-FCN model is proposed for analyzing the compressed data for semantic labelling of articles segments on news papers. Both the network architectures are explained in detail below.

1) Comp-HRNet: The Comp-HRNet architecture consists of three parallel convolution streams having resolutions $1/2$, $1/4$, and $1/8$ of input size. Since the DCT blocks in the input JPEG compressed stream contain both frequency(AC) and average spatial information (DC) coefficients, this type of arrangement extract the optimal hybrid representation by up-sampling and down-sampling data at different resolutions. First the input image passed through the stem block, as shown in detail in Fig-4(a), containing a convolution layer followed by Batch normalization, ReLU, and max-pooling layers sequentially to extract the optimized features from the compressed input. The input image F of size $256 \times 256 \times 3$ is convolved with 64 kernels of size 8×8 with stride one. Since the input stream is already in the form 8×8 blocks, The kernel of size 8×8 is selected to extract a meaningful compressed feature representation to make the further layers in the model comfortable. Next it goes through Max Pool layer with stride 2.

The input volume is transformed to $128 \times 128 \times 64$ which is then passed through 4 bottleneck residual blocks in sequence where each block contains a 1×1 , 3×3 and 1×1 convolution layers in sequence with stride of 1 and 64 output channels for the first two and 256 for the last as shown in Fig-4(b).After passing through the stem cell, an output of size $128 \times 128 \times 256$ is passed through two 3×3 convolution layer in parallel,one with stride of 1 and 32 output channels and other with stride of 2 and 64 output channels. The former output constitutes the $1/2$ resolution stream and the latter is the $1/4$ (down-sampled) resolution stream. Both these streams are parallely passed through 4 residual blocks in sequence(shown in Fig-3) which consists of a two 3×3 convolution layers of stride 1 and output channels same as that of input, followed by batch normalisation and ReLU as shown in Fig-4(c). The two streams are then fused by using 3×3 convolution layer with stride 1 and 2×2 upsampling for lower to higher resolution and 3×3 convolution of stride 2 for higher to lower resolution. During fusion, the output channels of the convolutions are equal to the channels in that specific stream. The $1/8$ resolution stream is made by passing the $1/4$ stream from the fused output through a 3×3 convolution of stride 2 and 128 output channels. These three streams are similarly passed through 4 sequential residual blocks in parallel. The outputs are then fused using 2×2 and 4×4 upsampling layers for $1/4$ and $1/8$ streams respectively, to finally create one stream of size $128 \times 128 \times 32$ which is passed sequentially through 3×3 convolution layer with stride 1, 2×2 upsampling layer, 1×1 convolution layer with stride 1 and n output channels, and linear sigmoid classifier to get the final output of size $256 \times 256 \times n$, where n is the number of classes as shown in Fig-3. The n channels of the output contain masks for articles, headers, images and advertisements without including the background.

The stem block is used to reduce the resolution of the input image into half and extract the key features before parallel multi-dimensional processing. The bottleneck layer consists of 3 convolutional layers with the first two having one-fourth of the output filters. We designed the bottleneck layer to be simple with just 3 convolutions because there wasn't any improvement in the accuracy of the trained model when more convolutions were added and with only 2 convolutions the accuracy dropped significantly. Similarly for the standard residual block we used 2 convolutions.

2) Fully Convolutional Network: The JPEG compressed stream is the input to the Comp-FCN model to perform

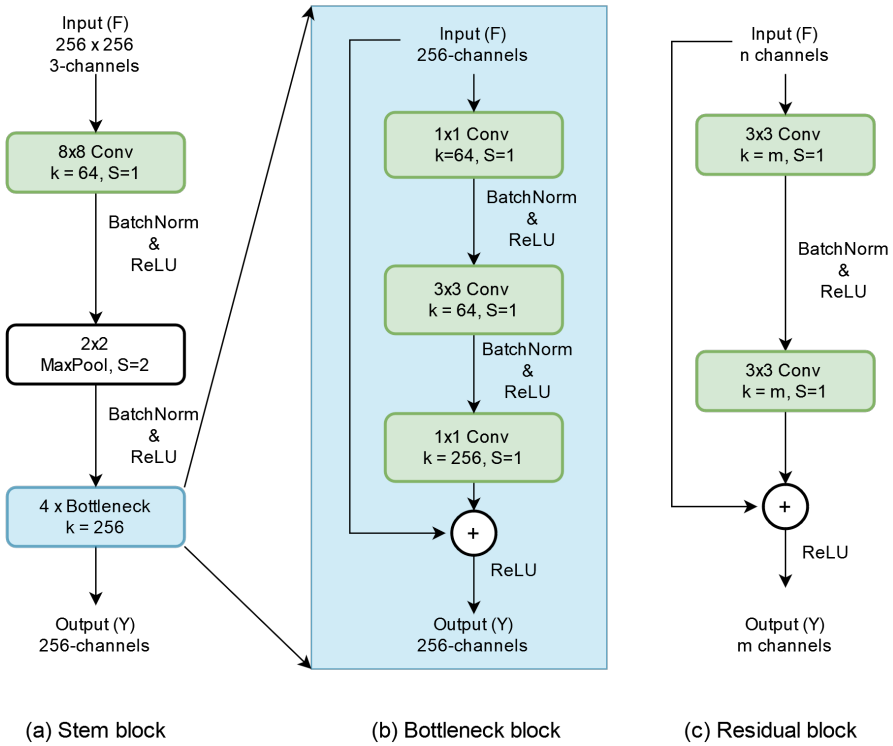


Fig. 4: The cross section view of (a) Stem block (b) Bottleneck block and (c) Residual block in the proposed Comp-HRNet architecture.

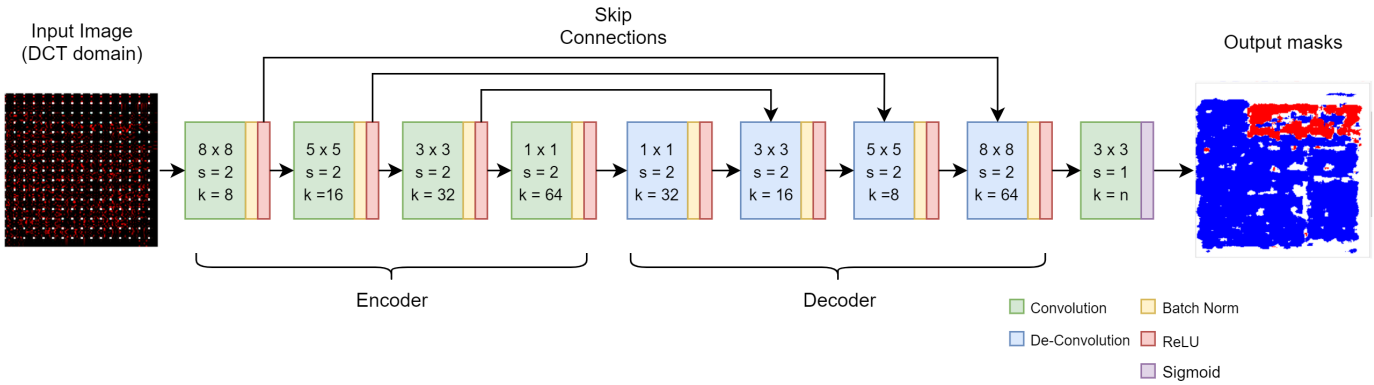


Fig. 5: The proposed Comp-FCN architecture for extracting the article segments in JPEG compressed news articles.

the segmentation directly on compressed image, and the architecture consists of an encoder and a decoder parts as shown in Fig-5. A stack of convolution layers are used in encoder part to extract and forward the optimal representation. Similarly, a stack of deconvolution layers are used in decoder to recover the semantic segments on the image. In specific, the compressed stream input to the encoder part consists of 4 convolution layers, where the first layer contains 8 number of kernel filters of size 8×8. And 16 number of filters with size 5×5 are used in second layer. In third layer 32 filters with kernal size 3×3 are used. Finally 64 filters with kernal size 1×1 are used in fourth layer. At each convolution layer the stride and padding of size 2 is set. Each convolutional layer is followed by a batch normalisation and a rectified linear unit (ReLU) layers to avoid the over fitting and include the non-linear feature activations. The optimal features extracted from the encoder part are fed into decoder part. Like encoder part, the decoder part also consists of 4 deconvolution layers, and rest of all the parameters are symmetrical to encoder part. The optimal features are fed to the decoder part where the deconvolution layers locate the semantic relation of each segment on the news paper. The skip connections are provided between first three convolution layer of the encoder and last three deconvolution layers in decoder part.

Since both models are fed with compressed input, we have used 8×8 filters at the beginning of the convolution layers to make model suitable to compressed input. This setting also help to carry the spatial information DC in each convolution of the image with kernel filters. The further experimental details with both the models are explained in the below section.

III. Experimental Results and Analysis

This section discusses the standard datasets, evaluation metrics, and experimental results for the proposed segmentation models.

A. Datasets

The proposed models are tested on two benchmark datasets. First one is Russian newspaper dataset consists of 101 images with ground truths for text regions marked with blue color, non-text with red color and the remaining pixels correspond to background. All the images have resolution of size 2400×3500 pixels and 300 dpi [44]. A sample DCT compressed input image, its ground truth with binary masks for the text and non-text regions are shown in Fig-6. The second dataset is a IIITA-ANA dataset developed by us. Since there are no open source datasets for recent news articles, we are motivated to develop this new dataset where all type of challenges are included. Here total 145 images are extracted from various recent printed newspapers in India. Each image is about 1024×512 in resolution. We annotated 4 major type of segments such as paper headers and headlines, text article, images, and other useful contents(Graphs, Charts, Advertisement) with colors Black, Red, Green, and Blue. The Images have been masked using GIMP, an open-sourced cross-platform image editor. This dataset will be made available online very soon. Since there are more number of objects and they appear multiple times in a typical news paper, the IIITA-ANA dataset is challenging to the proposed model. However, in this case both datasets are subjected to JPEG compression into DCT domain to feed the compressed representation into the proposed deep learning architecture.

B. Performance Metrics

In all experiments, the model's performance is evaluated by using three standard benchmark metrics - mean Intersection of Union(IoU), mean Dice coefficient and mean Average Precision (mAP) as provided in Eq. (18), (19) and (20) respectively. Where the Intersection of Union (IoU) is the fraction of true positives (TP) and sum of true positives(TP), false positives (FP) and false negatives(FN). This represents the ratio of area of overlap to the area of union. Similarly, Dice coefficient is the fraction of twice the area of overlap and total number of pixels in the predicted and ground truth masks. Mean Average Precision (mAP) gives information of the precision-recall curve as the weighted mean of precision achieved at 'n' threshold (P_n), with the increase in recall from the previous threshold$(R_n - R_{n-1})$ used as the weight.

$$Iou = \frac{TP}{TP + FP + FN} \tag{18}$$

$$DICE = \frac{2TP}{2TP + FP + FN} \tag{19}$$

$$mAP = \sum_{n} (R_n - R_{n-1}) P_n \tag{20}$$

Since it is a multi-class classifier problem, in each experiment the loss H of the model is calculated using categorical cross-entropy loss function as given in Eq. (21), where $\mathbf{y_i}$ is a ground truth mask and $\hat{y}_i$ is predicted mask.

$$H(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{y_i} \log_{\mathbf{e}}(\hat{\mathbf{y}}_\mathbf{i}) \tag{21}$$

C. Experimental Results

The proposed models are implemented using Tensorflow machine learning library and trained on a computational setup with GTX 1660ti 6GB GPU, where this computer has Intel i7-9750H CPU with 16GB RAM. For training, a batch size of 16 images and Adam optimizer with a learning rate of 0.001 are applied. The proposed models are separately trained and tested on the JPEG compressed version of the news articles in the two datasets. All the experiments are broadly divided into two types depending upon the dataset used as reported in Tables I and II. To show the generality of proposed models to work with pixel images and compare the computational efficacy with respect to experiments in compressed domain, a parallel experiment is conducted in uncompressed/pixel domain termed as Baseline approach. All the experiments and their corresponding results are discussed one by one as follows.

In first type, the proposed models are tested on JPEG compressed version of Russian dataset and the experimental results are shown in Table I. Here, the Comp-HRNET model is trained with 70% of the dataset for 100 epochs, and achieved 98.17% accuracy with 0.154 loss when tested on testing images. The model is further evaluated by Mean IoU, Mean DICE and mAP and the results are 82.76, 90.29 and 82.71 respectively. The following baseline approach with Comp-HRNET on Russian dataset is conducted and results are tabulated in same table. Likewise, the second model Comp-FCN model is trained for 100 epochs on 70% dataset and achieved 93.63% accuracy with 0.179 loss when tested. The Mean IoU, Mean DICE and mAP are 79.66, 88.34 and 82.20 respectively. Even here the baseline approach results in uncompressed domain are calculated as shown in table. In all the experiments, the methods have achieved state of the art performance in compressed domain and achieved the performance close to the baseline approach in uncompressed domain [4]. For both the models, the training performance, the accuracy and loss per each epoch are plotted as shown in the Fig-7(a) and Fig-7(b). In the figures it is observed that the models are saturated at 100 epochs. However, the news articles in Russian dataset were old and most of articles contains background noise and black and white in appearance, and are very few in number. Because of these reasons the segment boundaries on the DCT representation was not clear and the performance on this representation is reduced when compared to baseline model. Therefore we have tested the proposed models on the recent news articles in IIITA-ANA dataset as explained below.

In second type, the Comp-HRNet model is trained on JPEG compressed version of IIITA-ANA dataset. The model is trained for 100 epochs with 70% of the dataset, and achieved the 98.86% performance with cross entropy loss 0.05 when tested. The corresponding Mean IoU, Mean DICE and mAP are 93.20, 96.02, and 79.22 are shown in Table II. The

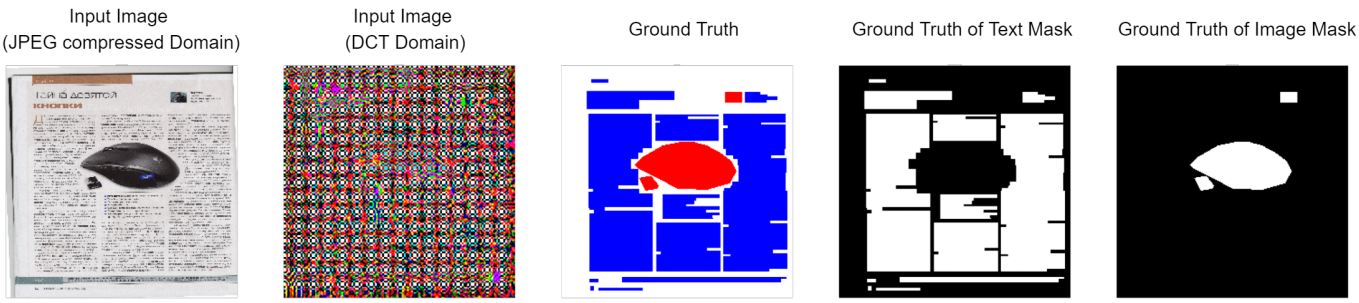
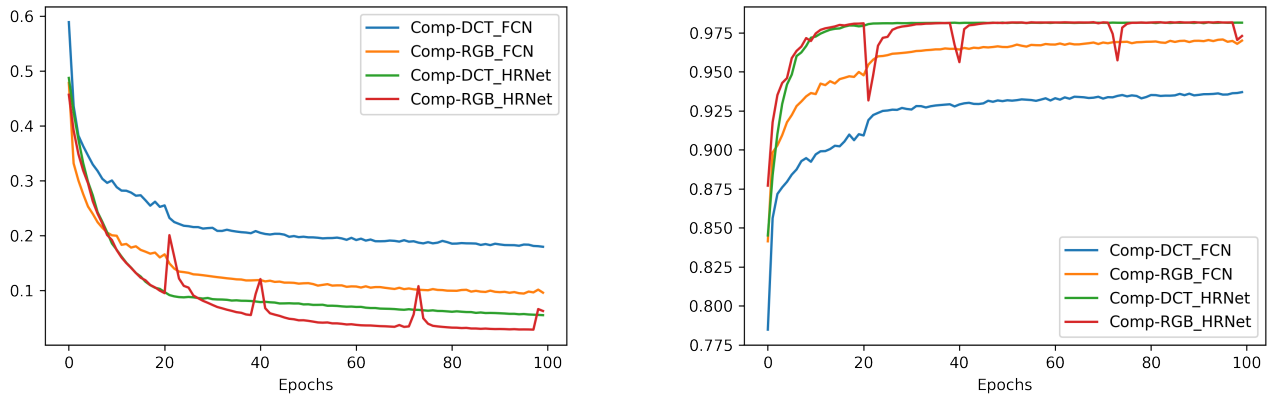


Fig. 6: A sample input DCT stream extracted from JPEG compressed news articles of Russian dataset and the corresponding ground truths for text and non-text segments.

TABLE I: The experimental performance of proposed model tested on the JPEG compressed version of the Russian Dataset

Model Type	Accuracy (%)	Cross Entropy Loss	Mean IoU	Mean DICE	mAP
Comp-HRNet(DCT Images)	98.17	0.154	82.76	90.29	82.71
Comp-HRNet(Pixel images) Baseline Approach	98.17	0.085	95.72	97.79	93.57
Comp-FCN (DCT Images)	93.63	0.179	79.66	88.34	82.20
Comp-FCN (Pixel Images) Baseline Approach	97.06	0.0854	91.07	94.70	91.4



(a) Plot of Loss per epoch

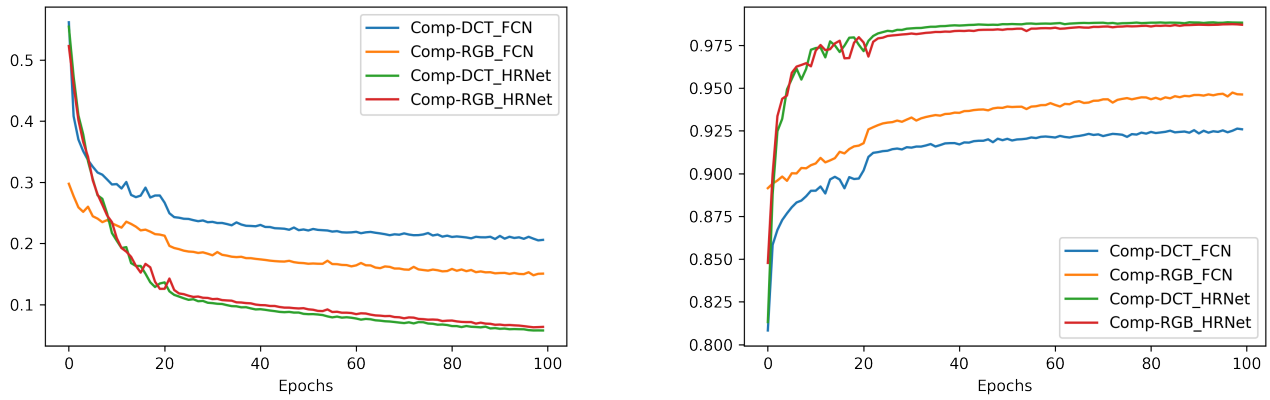
(b) Plot of Accuracy per epoch

Fig. 7: The training accuracy and lose per each epoch of models experimented on the Russian Dataset

TABLE II: Experimental results of the proposed two deep learning models tested on the JPEG compressed version of the IIITA-ANA dataset

Model Type	Accuracy (%)	Cross Entropy Loss	Mean IoU	Mean DICE	mAP
Comp-HRNet (DCT Images)	98.86	0.05	93.20	96.02	79.22
Comp-HRNet(Pixel images) Baseline Approach	98.60	0.0667	93.36	96.22	81.23
Comp-FCN (DCT Images)	92.94	0.1809	49.04	59.88	77.07
Comp-FCN (Pixel Images) Baseline Approach	94.59	0.09	61.33	67.56	79.20

following baseline approach for Comp-HRNet is trained and tested on IIITA-ANA dataset, and the results are tabulated in Table II. It is observed that the proposed model has achieved performance close to the baseline model. Similarly, the Comp-FCN model is trained for 100 epochs on JPEG compressed IIITA-ANA dataset, and achieved 92.94% accuracy with 0.18 loss when tested. The Mean IoU, Mean DICE and mAP are 49.04, 59.88 and 77.07. The corresponding baseline approach on Comp-FCN is conducted, the results are shown in table II. The accuracy and loss for each epoch is plotted as shown in the Fig-8(a) and Fig-8(b). Some of the semantic segmentation results of both models tested on JPEG compressed IIITA-



(a) Plot of Loss per epoch

(b) Plot of Accuracy per epoch

Fig. 8: The training accuracy and lose per each epoch of models experimented on IIITA-ANA Dataset

TABLE III: The experimental performance of proposed model tested on the JPEG compressed version of the Publay Dataset

Model Type	Accuracy (%)	Cross Entropy Loss	Mean IoU	Mean DICE	mAP
Comp-HRNet(DCT Images)	97.15	0.1709	80.44	89.15	81.91
Comp-HRNet(Pixel images) Baseline Approach	97.84	0.1728	78.49	87.93	83.47
Comp-FCN (DCT Images)	92.17	0.5059	62.26	76.68	78.20
Comp-FCN (Pixel Images) Baseline Approach	94.49	0.3707	70.87	82.87	79.32

ANA dataset are shown in Fig-9. Compared to the Comp-HRNet model the Comp-FCN model's performance is low in segmenting the segment boundaries smoothly. It is because of loosing some fine details for complex image segments when they go through the upsampling and downsampling stages. But the model performed well on Russian dataset as it has to simply differentiate the text from non text parts. however, overall, it is noted that in all the experiments the proposed Comp-HRNET and Comp-FCN methods have achieved the closer performance with reduced computational costs compared to the traditional baseline approach.

The experiments are conducted to calculate the class wise performance of the proposed models tested on JPEG compressed IIITA-ANA dataset as shown in Table III. The Comp-HRNET model has achieved more than 97% of accuracy and the Comp-FCN model has achieved more than 91% accuracy in average for all the classes. Since JPEG algorithm also uses the 4×4 block division in some cases, further, the experiments are conducted based on JPEG compressed 4×4 DCT block representation of the IIITA-ANA dataset. Both models are trained and tested with this type of input, and the corresponding experimental results are shown in Table IV. The models showed the similar performance as compared to 8×8 DCT block representation. Further, the robustness of the model is analyzed on the noisy input, where Salt and Pepper noise of size 0.02% is added to the input image. It is noticed that the model could reasonably locate the object segments in the JPEG compressed news paper as shown in Fig-11

The present technology has achieved the significant progress in reducing the computation costs. However, in present Big data era, a small reduction of time $1ms$ per an image shall make difference in many more hours of time when a million images and even larger volumes are concerned. Therefore, with this motivation, in order to show how the computational time gain could be possible in JPEG compressed domain, the time taken for each operation is measured for Baseline JPEG encoding and decoding stages for 5 random images from the IIITA-ANA dataset as shown in Fig-12. All the operations in encoding and decoding stages have been manually developed by us using python language, and the source code is provided in Github [45]. The details of those measurements in average are tabulated as shown in the TABLE VI. From the table, it is noticed that, during encoding stage the block encoding stage which includes DCT and Quantization has taken nearly 50% portion of the total encoding time. The other operations have taken the rest of the time. Similarly, the memory variation for each operation in encoding and decoding stage are calculated as shown in TABLE VI. Similarly when memory is concerned, it is noticed that the size got increased after DCT transformation stage. It is because the pixels of integer type become signed float values to accommodate the DCT coefficients, the memory size of the block decoding stage became high as shown in TABLE VI. However, during decompression stage the computational costs and memory size for each stage are nearly same with slight variation. In the table, it is noticed that the computational time and memory are varying. Therefore it

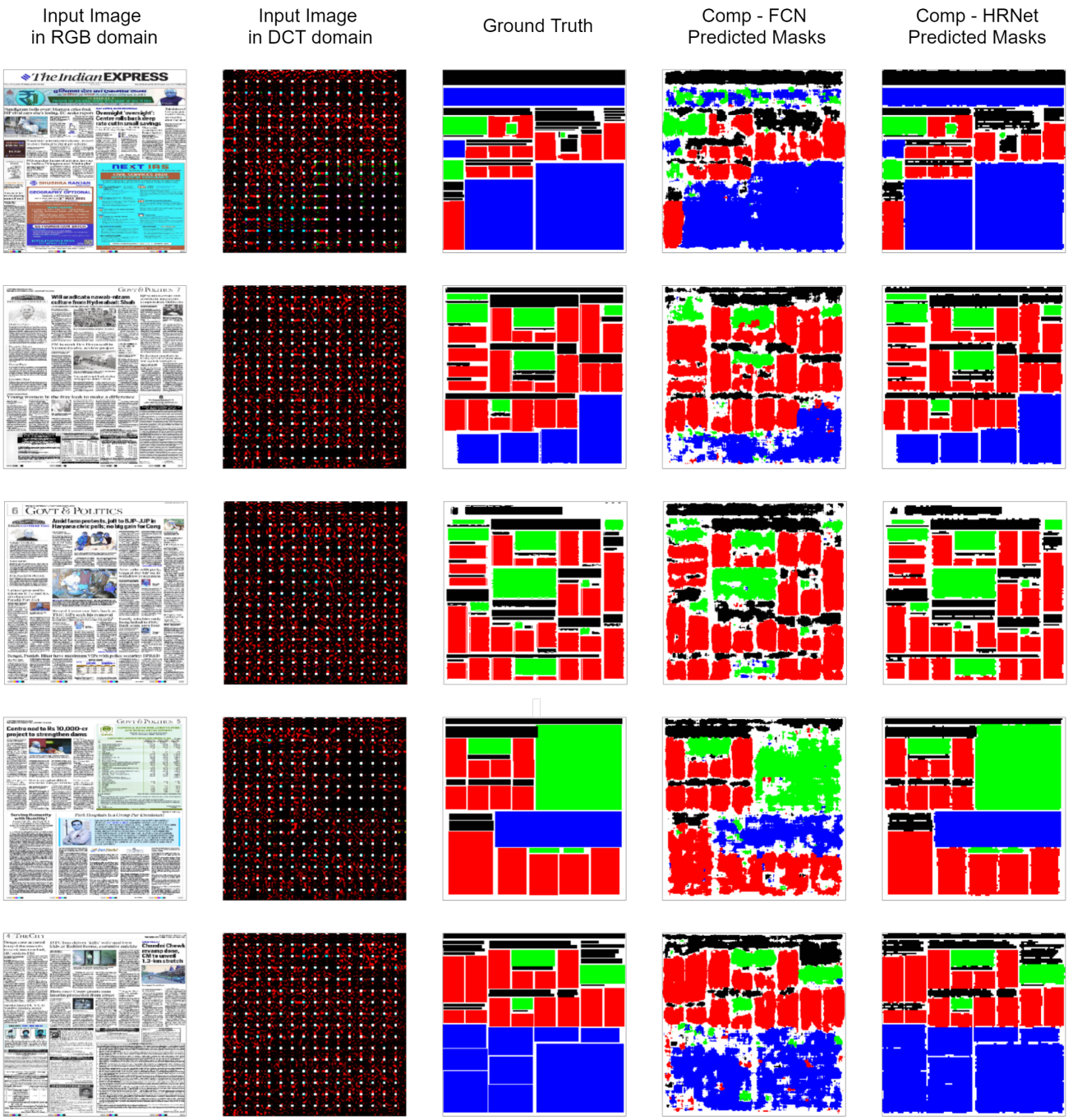


Fig. 9: The sample output images with predicted labels of news articles of the proposed models Comp-FCN and Comp-HRNet tested on IIITA-ANA dataset

TABLE IV: The classwise performance analysis of the Comp-HRNet and Comp-FCN models tested on the IIITA-ANA Dataset

Model	**Class Type**			
	Articles (%)	**Headers (%)**	**Advertisements (%)**	**Images (%)**
Comp-HRNet	97.97	98.34	99.70	99.45
Comp-FCN	91.39	91.20	95.41	93.76

is better to provide a generic method to show how we can achieve the gain in speed and size as explained below.

In order to measure the the computational gain and reduced memory costs, let the total computational complexity C_u in uncompressed domain be $T_u = DeCompCost + N \times Operation + comCosts$, where N is the total number of pixel

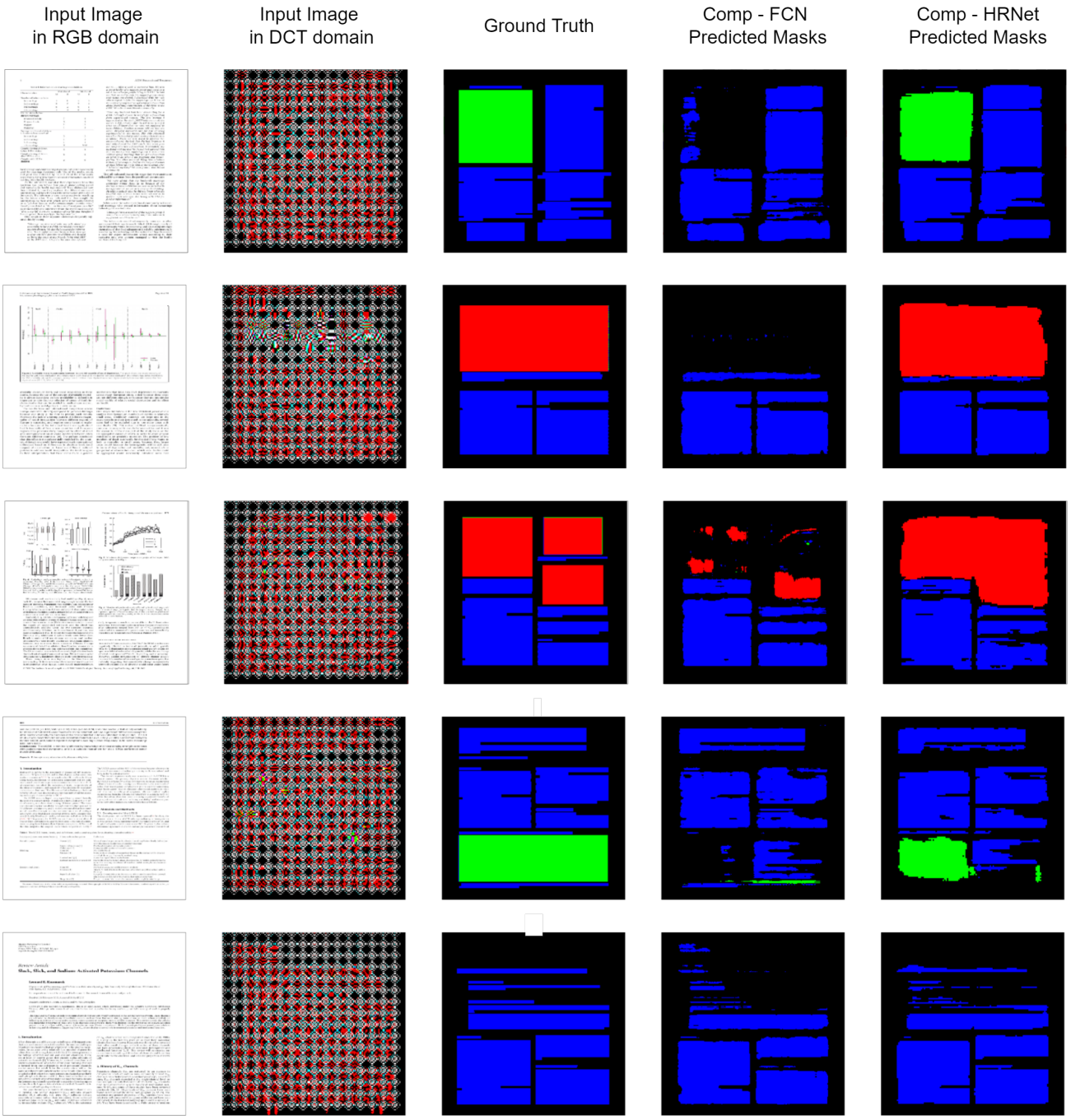


Fig. 10: The sample output images with predicted labels of news articles of the proposed models Comp-FCN and Comp-HRNet tested on Publay dataset

TABLE V: Experimental results of the proposed models tested on the (4×4) DCT compressed JPEG compressed IIITA-ANA dataset.

Model Type	Input Type	Accuracy (%)	Cross Entropy Loss	Mean IoU	Mean DICE	mAP
Russian Dataset	Comp-HRNet	86.09	0.454	64.43	74.42	76.38
	Comp-FCN	89.7	0.5091	61.08	71.99	76.47
IIITA-ANA Dataset	Comp-HRNet	92.89	0.2289	55.26	65.01	79.74
	Comp-FCN	90.11	0.2728	36.76	47.85	77.86

in the compressed news article input image, $DeCompCost$ is costs involved for decompression of the compressed images, $operation$ is amount of time for any typical operation on it, and $compCost$ is costs to re-compress that image. Whereas,

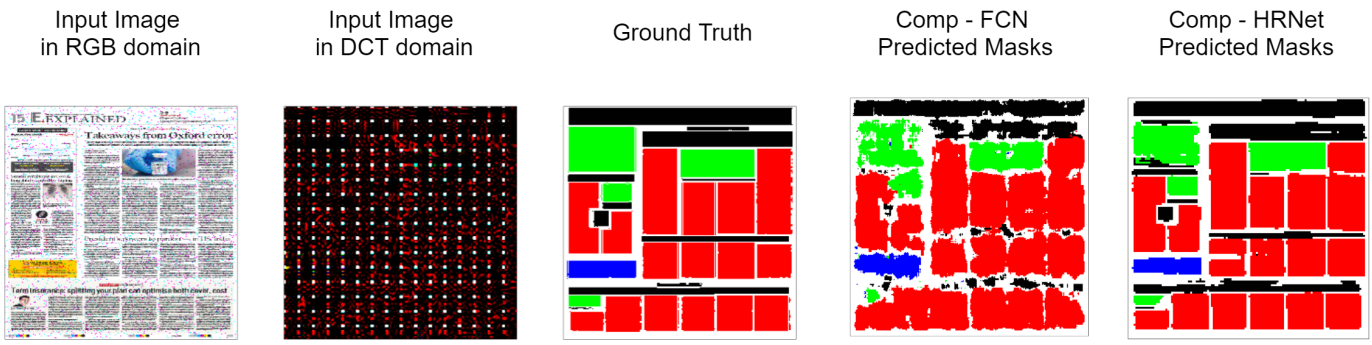


Fig. 11: Predicted masks of the IIITA-ANA Dataset in Pixel domain trained using Comp-FCN and Comp-HRNet using a Salt and Pepper Noisy input(amount 0.02)

Fig. 12: The sample news articles from the IIITA-ANA dataset used to apply JPEG compression algorithm to analyze the variation of computational time and memory for each stage during compression and decompression.

in case of feeding direct JPEG stream, only a partial decompression is required. Where, the decoding of haffmann, run length and zigzag stepts are applied. feeding the representation after this partial decompression involves the 45% and avoids 55% of total decompression time per an image. For more understanding, in conventional approach the total time T_U required is U units of time in which decompression share $x\%$, operation time $y\%$ and recompression time $z\%$. Where as performing the same task on compressed representation requires the $(1/2)x\%$ decompression time and $y\%$ of operation time and 0% of recompression time. It means it requires $1/2(x) + y + 0\%$ of time which equals to 48.3% and avoids $1-(1/2(x)+y+0)\%$ of time which equals to 51.7% of time. this reduction in time achieves the computational gain twice the conventional approach.

The performance of the proposed models with baseline approach are compared with recent state-of-the-art methods for segmentation task in the uncompressed domain as shown in Table VII. It is noticed that the proposed models have achieved best performance with reduced costs. Similarly the proposed models performance is compared with state-of-the-art methods directly in JPEG compressed domain. It is noticed that the models have improved the accuracy significantly for segmentation task directly on JPEG compressed news papers as reported in Table VIII. Finally, it is anticipated that the models with compressed data as input shall redefine the many research methods in terms of computational complexities in future.

IV. Conclusion

This paper has proposed two deep learning models, Comp-HRNet and Comp-FCN, for segmenting the news articles in JPEG compressed news papers. The JPEG stream is obtained by applying the partial decompression on JPEG compressed news articles. The DCT stream is directly fed as input to the proposed models. The model is trained and tested on two benchmark datasets. The experimental results show that the proposed models have achieved significant performance with reduced computational costs in segmenting the news articles directly in the JPEG compressed domain. Similarly, the same model has outperformed the recent methods with state-of-the-art accuracy in the pixel domain when trained on pixel domain. The models have been analyzed with different evaluation metrics. Exploring solutions to various DIA problems directly in the JPEG compressed domain is our future work.

V. Acknowledgments

This work is supported by the Ministry of Education (formerly MHRD), Government of India. The authors are thankful to CVBL Lab at IIIT, Allahabad, for providing the computer facility to carry out the experiments. The authors also thank the Google Colab platform for providing a free GPU computation facility for training and testing the proposed methods.

References

[1] I. Library, "Newspapers and Magazines as Primary Sources." https://guides.library.illinois.edu/c.php?g=593567p=4105853, 26 Jan 2021. [Online; accessed 17-Aug-2021].

[2] T. Mills, "Preserving yesterday's news for today's historian: a brief history of newspaper preservation, bibliography, and indexing," *The Journal of Library History (1974-1987)*, vol. 16, no. 3, pp. 463–487, 1981.

[3] A. Bansal, S. Chaudhury, S. D. Roy, and J. Srivastava, "Newspaper article extraction using hierarchical fixed point model," in *2014 11th IAPR International Workshop on Document Analysis Systems*, pp. 257–261, IEEE, 2014.

[4] A. Almutairi and M. Almashan, "Instance segmentation of newspaper elements using mask r-cnn," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 1371–1375, IEEE, 2019.

[5] LETTER.LY, "Online Web Page." https://letter.ly/newspaper-statistics/, 2021. [Online; accessed 09-Aug-2021].

TABLE VI: The detailed analysis of encoding and decoding times required for each sub-operation in JPEG algorithm applied on a typical RGB image of size 1728Kb and 1024×1024, and the variation of sizes at each corresponding operation.

Operation	**JPEG-Encoding**		**Operation**	**JPEG-Decoding**	
	%Time	**%Space**		**%Time**	**%Space**
YCbCr	0.13%	1728Kb	**Run length + Huffmann**	12.64	6912Kb
DCT + Quant	49.64%	6912Kb	**Zigzag**	32.15	6912Kb
Zigzag	31.45%	6912Kb	**DCT + Quant**	55.17	1728Kb
Run length + Huffmann	18.76%	179Kb	**YCbCr-RGB**	0.02	1728Kb

TABLE VII: Comparing the performance of the proposed model tested on Russian dataset with the results of recent models in uncompressed domain (Pixel images).

S.No	**Year**	**Model**	**mAP**
1	2014	SVM [46]	76.31%
2	2017	FCN [25]	-
3	2019	mask R-CNN [4]	81.6%
4	-	Comp-HRNet	93.57%
5	-	Comp-FCN	91.4%

TABLE VIII: Comparing the performance of the proposed model tested on Russian dataset in JPEG compressed domain with the results of recent models in compressed domain .

S.No	**Year**	**Model**	**mAP**
1	2019	DCT-EDANet [43]	61.6
2	-	Comp-HRNet	82.71%
3	-	Comp-FCN	82.50%

[6] C. Wick and F. Puppe, "Fully convolutional neural networks for page segmentation of historical document images," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pp. 287–292, 2018.

[7] S. Mandal, S. Chowdhury, A. Das, and B. Chanda, "Automated detection and segmentation of table of contents page from document images," in *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pp. 398–402 vol.1, 2003.

[8] D. Reuse, "Digitizing Data Using Optical Character Recognition (OCR)." https://www.design-reuse.com/industryexpertblogs/48149/digitizing-data-using-optical-character-recognition-ocr.html, 15 Jan 2020. [Online; accessed 17-Aug-2021].

[9] A. Antonacopoulos and R. Ritchings, "Segmentation and classification of document images," in *IEE Colloquium on Document Image Processing and Multimedia Environments*, pp. 16/1–16/7, 1995.

[10] R. S. Medeiros, J. Scharcanski, and A. Wong, "Natural scene segmentation based on a stochastic texture region merging approach," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1464–1467, 2013.

[11] A. N. Kumar, M. Ilamathi, C. Jothilakshmi, and S. Kalaiselvi, "Outdoor scene image segmentgation using statistical region merging," in *2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering*, pp. 351–354, 2013.

[12] C. Fowlkes, D. Martin, and J. Malik, "Learning affinity functions for image segmentation: combining patch-based and gradient-based approaches," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 2, pp. II–54, 2003.

[13] Z. Shi and V. Govindaraju, "Multi-scale techniques for document page segmentation," in *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, pp. 1020–1024 Vol. 2, 2005.

[14] F. Zirari, A. Ennaji, S. Nicolas, and D. Mammass, "A document image segmentation system using analysis of connected components," in *2013 12th International Conference on Document Analysis and Recognition*, pp. 753–757, 2013.

[15] K. Y. Wong, R. G. Casey, and F. M. Wahl, "Document analysis system," *IBM Journal of Research and Development*, vol. 26, no. 6, pp. 647–656, 1982.

[16] K. Hadjar and R. Ingold, "Arabic newspaper page segmentation," in *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pp. 895–899, 2003.

[17] K. Chaudhury, A. Jain, S. Thirthala, V. Sahasranaman, S. Saxena, and S. Mahalingam, "Google newspaper search – image processing and analysis pipeline," in *2009 10th International Conference on Document Analysis and Recognition*, pp. 621–625, 2009.

[18] R. Elanwar, W. Qin, and M. Betke, "Making scanned arabic documents machine accessible using an ensemble of svm classifiers," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 21, no. 1, pp. 59–75, 2018.

[19] K. Hadjar and R. Ingold, "Logical labeling of arabic newspapers using artificial neural nets," in *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, pp. 426–430, IEEE, 2005.

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[21] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[22] S.-Y. Lo, H.-M. Hang, S.-W. Chan, and J.-J. Lin, "Efficient dense modules of asymmetric convolution for real-time semantic segmentation," in *Proceedings of the ACM Multimedia Asia*, pp. 1–6, 2019.

[23] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," 2020.

[24] H. Wu, C. Liang, M. Liu, and Z. Wen, "Optimized hrnet for image semantic segmentation," *Expert Systems with Applications*, vol. 174, p. 114532, 2021.

[25] B. Meier, T. Stadelmann, J. Stampfli, M. Arnold, and M. Cieliebak, "Fully convolutional neural networks for newspaper article segmentation," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 414–419, IEEE, 2017.

[26] M. Javed, P. Nagabhushan, and B. B. Chaudhuri, "A review on document image analysis techniques directly in the compressed domain," *Artificial Intelligence Review*, vol. 50, no. 4, pp. 539–568, 2018.

[27] M. Ulicny and R. Dahyot, "On using CNN with dct based image data," in *Proceedings of the 19th Irish Machine Vision and Image Processing conference IMVIP*, 2017.

[28] M. Pistono, G. Coatrieux, J. Nunes, and M. Cozic, "Training Machine Learning on JPEG Compressed Images," in *2020 Data Compression Conference (DCC)*, pp. 388–388, 2020.

[29] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[30] C. Florea, M. Gordan, B. Orza, and A. Vlaicu, "Compressed domain computationally efficient processing scheme for jpeg image filtering," in *Advanced Engineering Forum*, vol. 8, pp. 480–489, Trans Tech Publ, 2013.

[31] B. Shen and I. K. Sethi, "Direct feature extraction from compressed images," in *Storage and Retrieval for Still Image and Video Databases IV*, vol. 2670, pp. 404–415, International Society for Optics and Photonics, 1996.

[32] B. Shen and I. K. Sethi, "Convolution-based edge detection for image/video in block DCT domain," *Journal of Visual Communication and Image Representation*, vol. 7, no. 4, pp. 411–423, 1996.

[33] R. L. de Queiroz and R. Eschbach, "Fast segmentation of the JPEG-compressed documents," *Journal of Electronic Imaging*, vol. 7, no. 2, pp. 367–378, 1998.

[34] Y. Lu and C. L. Tan, "Document retrieval from compressed images," *Pattern Recognition*, vol. 36, no. 4, pp. 987–996, 2003.

[35] M. Javed, P. Nagabhushan, B. B. Chaudhuri, and S. K. Singh, "Edge based enhancement of retinal images using an efficient jpeg-compressed domain technique," *Journal of Intelligent & Fuzzy Systems*, no. Preprint, pp. 1–17, 2019.
[36] L. Gueguen, A. Sergeev, B. Kadlec, R. Liu, and J. Yosinski, "Faster neural networks straight from JPEG," in *Advances in Neural Information Processing Systems*, pp. 3933–3944, 2018.
[37] B. Rajesh, M. Javed, Ratnesh, and S. Srivastava, "DCT-CompCNN: A Novel Image Classification Network Using JPEG Compressed DCT Coefficients," in *2019 IEEE Conference on Information and Communication Technology*, pp. 1–6, 2019.
[38] M. K. Khandani and W. B. Mikhael, "Training strategies for convolutional neural networks with transformed input," in *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1058–1061, IEEE, 2021.
[39] B. Deguerre, C. Chatelain, and G. Gasso, "Fast object detection in compressed JPEG Images," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 333–338, 2019.
[40] B. Rajesh, M. Javed, and P. Nagabhushan, "Automatic Tracing and Extraction of Text-Line and Word Segments Directly in JPEG Compressed Document Images," *IET Image Processing*, 04 2020.
[41] B. Rajesh, P. Jain, M. Javed, and D. Doermann, "HH-CompWordNet: Holistic Handwritten Word Recognition in the Compressed Domain," in *2021 Data Compression Conference (DCC)*, 2021.
[42] M. Ehrlich and L. Davis, "Deep Residual Learning in the JPEG Transform Domain," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3483–3492, 2019.
[43] S.-Y. Lo and H.-M. Hang, "Exploring semantic segmentation on the dct representation," in *Proceedings of the ACM Multimedia Asia*, pp. 1–6, 2019.
[44] A. Vilkin and I.Safonov, "UCI Machine Learning Repository."
[45] A. Kumar, "JPEG-Compression-algorithm." https://github.com/AnandK27/jpeg-encoding, 2021. [Online; accessed 30-Oct-2021].
[46] A. Bansal, S. Chaudhury, S. D. Roy, and J. Srivastava, "Newspaper article extraction using hierarchical fixed point model," in *2014 11th IAPR International Workshop on Document Analysis Systems*, pp. 257–261, 2014.