International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V69I10P229

Volume 69 Issue 10, 227-234, October, 2021
©2021 Seventh Sense Research Group®

Memory Enhanced Dynamic Conditional Random Fields Embedded Pairwise Potential CNN for Fabric Defects Identification

B. Vinothini[1], S. Sheeja[2]

[1]*Research Scholar, Department of CS, CA & IT, Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India.*
[2]*Professor, Department of CS, CA & IT, Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India.*

[1]vinothini.prabhu@gmail.com, [2] sheejaajize@gmail.com

Abstract — *In the textile industry, defect identification is the most crucial process for localizing Fabric Defects (FDs) and enhancing yarn quality. In earlier centuries, many techniques were discussed to identify the FDs automatically. Among those, a hybrid technique called Pairwise-Potential Activation Layer in Convolutional Neural Network (PPAL-CNN) localizes the fine structures in textile imagery through integrating dynamic AL on CNN and a PP factor in the Conditional Random Fields (CRFs). However, this CRF should be supplied a priori rather than learned. It was hard for a complex interaction between FDlabels/classes while performing multiple, or long-range dependencies exist. Therefore, this paper proposes an Enhanced PPAL-CNN (EPPAL-CNN) technique to manage the complex structure interaction of FDs. First, the CRF is extended by integrating external memory strategies stimulated from the memory networks and thus facilitating CRFs for interpretation beyond localized characteristics and have access to the complete image. It encompasses the memory and Dynamic CRF (DCRF) layers. The memory layer is partitioned into input, output, and current input memory. The interpretations of input and output memory have interacted through an attention model in which weights are calculated by the relation of an input and a current input memory. Then, an outcome of the memory layer is taken as input to the DCRF layer. The DCRFs are simplified linear CRFs and are used for defining the shared hidden state and complicated relation between labels. Its factorial construction includes the relations among cotemporally labels, explicitly modeling constrained likelihood dependencies among various labels. So, a high-level Markov dependency among labels is modeled by considering the external memory. Finally, the investigational outcomes exhibit that the EPPAL-CNN achieves 93.36% accuracy compared to the PPAL-CNN technique using the TILDA database.*

Keywords — *Fabric defects, Defect identification, CNN, Pairwise-potential activation, CRFs, Memory network.*

I. INTRODUCTION

The manufacturing of textiles is a frequently practiced regular commodity. Usually, a natural ingredient is applied to create textile fibers. The design process reveals a defect in the material's foundation. A faulty steering system or the fabric's breakage on the sewing machine may cause a structural difference between the duration of its discovery in filament, weft, or spot flaws like the misdrawing of the brace, infrastructure, inconsistency, and slub. Faults can reduce manufacturing costs by 45-65%. Weavers can inspect the textile material for highly complex flaws in classical Sewing Machines by periodically crossing a couple of machines, as every fiber fault can be prevented or remedied when detected [1].

So, the fashion domain is advanced towards the automated monitoring of textiles for specific fabric consistency measurements. Automation is a performance measurement method, which recognizes and informs defects in production goods, often called assessments. Commonly, manual testing on wood panels is the only way of ensuring consistency and helping to repair minor flaws on a timely basis. But at the same time, strain triggers a systematic error, and minor deficiencies are always undiscovered. Detection accuracy can be exploited by the usual automated fabric analysis of about 80% than the manual assessment [2]. Thus, automated assessments are a natural solution to grow fabric productivity by reducing manufacturing costs. However, this is hard at times. Several automated fiber analysis techniques depend on machine vision tools, namely image analysis and data mining schemes which can segment and recognize the defective fabrics. The FDs identification techniques are categorized into numerical, contextual, hierarchical, composite, training, and model-based techniques. These techniques are sensitive to errors, high cost, limited to special flaws and conflicting with changes in fiber quality and context. During the past few years, many schemes have been developed for recognizing FDs.

Among those techniques, the hybrid algorithm is designed to achieve a high sturdiness in controlling dissimilarities in fabric patterns and defect categories. But they were unsuccessful in recognizing flaws when compared to the repetitive unit of a patterned fiber. In recent centuries, deep learning such as CNN has been adopted for the effective segmentation of textile imagery. The types of CNNs include Fully Convolutional Network (FCN) [3], U-Net [4], SegNet [5], and so on, each distribute the fundamental units such as convolution, pooling, and activation processes where pooling is considered to prevent overfitting and lessen the spatial dimensions. But, these offer features along with an overall semantic significance and intellectual data, which are not appropriate for segmenting adequate image details since standard CNNs are highly accessible, and small patterns are minimized via pooling [6].

So, many FDs are taken into account as small structures as they are denoted only via a less amount of image pixels. A flawed look on textile imagery like dual, loss, rough picks, and so on indicates a small/adequate surface pattern and always includes smaller than 35% of pixels providing an imbalanced FD database. To enhance the small structure segmentation, an additional task is essential to modify CNN's coarse outcomes. The other usual challenge in the deep learning algorithms is that data from real-time applications might not often uniformly share among classes [7]. So, two essential functions are accounted for constructing the CNN for identifying the FDs: maintenance of small patterns and handling an imbalanced database. The required CNN must not comprise several convolutional layers for avoiding data on textile imagery from missing and must prevent by pooling for retaining the image perseverance in the feature maps.

From this perspective, the PPAL-CNN technique [8] was developed, which adopts statistical flaw details and a CNN for identifying the FDs. Originally, the motif of fabrics was computed via the auto-correlation of fiber images for interpreting the recurring fabric textures. After, a motif-center-point map was created via normalizing the cross-correlation. The node point distributions can specify the fabric texture's stability to derive the statistical rule. This statistical rule was applied as PPAL in CNN for associating the node points in a motif area to the flaw decision. Moreover, a defect likelihood map was applied as a CNN's dynamic ALalong withCRF's PP factor for accurately localizing small structures and handling the imbalanced dataset during CNN training.

But, this CRF should be supplied a priori rather than learned. This was difficult for complicated relations among FD classes when executing multiple or long-range dependences occur. Hence, this article designed the san EPPAL-CNN technique for handling the complex structure interaction of FDs. Initially, the CRF is extended via combining external memory strategies inspired by the memory networks and so allowing CRFs to indicate beyond localized features and utilize the whole image. It encompasses memory and DCRF layers. The memory layer is split into input, output, and the current input memory, i.e., a present stage. The interpretations of input and output memory are associated through attention which determines the weights using the relevance of input and current input memory. Then, the result of the memory layer is considered as input to the DCRF layer, which is a simplification of linear CRFs. Its factorial construction includes the connections among cotemporally labels, clearly forming restricted likelihood dependences among different labels. This creates a higher-order Markov dependence between labels via accounting an external memory. Thus, it improves the efficiency of PPAL-CNN for identifying the complex related FDs efficiently.

The following sections are included: Section II surveys the researches related to the identification of FDs. Section III explains the methodology of the EPPAL-CNN technique, and Section IV discusses its performance. Section V summarizes this research work and suggests future scope.

II. LITERATURE SURVEY

A context-awareness and local texture saliency-based FD detection [9] has been suggested, which applies Local Binary Pattern (LBP), salient region identification, and segmentation using an optimum threshold. Initially, a target image was partitioned into segments, and the LBP method was employed for extracting the texture features of segments. Then, many other segments were selected randomly for determining the LBP contrast between a given block and the randomly selected blocks. Also, a saliency map was created depending on the determined contrast data. Further, the saliency map was segmented by an optimum threshold obtained via an iterative method for identifying the FDs. But it has high computational complexity.

A discriminative interpretation [10] was recommended for detecting the patterned FDs. Primarily, fabric images were partitioned into similar size patches. Then, the learning of Fisher Criterion-based Stacked Denoising Auto-Encoder (FCSDAE) was executed to distinguish both defect and defect-free samples. Further, the thresholding scheme was used for calculating a residual factor of recreated and defect images. But, its false alarm rate was slightly high.

An improved method using the adaptive K-means algorithm [11] was suggested, which produces lattice segments and templates for recognizing FDs in an automated way. In this method, the texture primitive principles assembled in individual texture classes were split into lattices via considering textile imagery standards. For every texture class, different pre-inspection efficacy analyses were carried out on a variety of defect-less images. Also, a template mapping was described for a lattice segmented from an image, and the lattices with distances greater were compared to the trained distance were recognized to be defective. But the runtime was high because of the fiber structure dissimilarities.

An automated learning-based method [12] was proposed for detecting FDs depending on the interpretation of fiber structure via the Redundant Contourlet Transform (RCT). First, pre-processing was used to identify the fundamental structure dimension and decompose the images. After, categorize fiber images were trained by the Bayes classification for distinguishing flaws and perfect fibers. But it doesn't give correct outcomes in a few situations where there exists dependency among variables.

An artificial learning method [13] was suggested for recognizing and localizingFDs. In this method, a Multi-Scale Convolutional Denoising Auto-Encoder (MSCDAE) was employed for recovering the image slices and synthesizing the impacts from the relevant channels. The residue of every image patch was considered because of the prediction of the clear pixel. At every point of resolution, the residual map was segmented and synthesized for obtaining the outcome. But, it was trained only on some defect-free samples and was more sophisticated because of high computation time.

A Multi-scaling CNN (MCNN) [14] was designed to recognize FDs where outcomes of every CNN were averaged via an overlapping averaging scheme for increasing the accuracy. Also, a classical Alex Netframework was applied for learning MCNN by taking into consideration of the TILDA database. However, the mean error rate was high. A novel Defect Enhancement Generative Adversarial Network (DEGAN) [15] framework was developed depending on the defect improvement algorithm in a forward channel and after discriminator. A generation and discriminant model were designed to reconstruct the actual defect data and retain the defect characteristics. Moreover, a specified reconstruction error method was used for attributes with microcrack defects to create more detailed local defect characteristics. But the hyper-parameters used in this DEGAN were needed to optimize for enhancing efficiency.

An enhanced unsupervised model [16] was developed to identify FDs via Deep Convolutional GAN (DCGAN) and reconstruct the fabrics maintained in the query image. Here, the defects were identified in the background via creating a residual map during reconstruction. Also, the discriminator was used to enhance the efficacy. But, average accuracy was not effective. An efficient CNN called Mobile-UNET [17] was structured for segmenting flaws using MobileNetV2 feature extractor and 5 deconvolution units. The mid-band matching loss function and a depth-wise independent convolution were employed to resolve the data imbalance issue and difficulty. At last, SoftMax was employed for creating a segmented mask. But, it was not apt for less data for training.

III. PROPOSED METHODOLOGY

This part describes the EPPAL-CNN technique in detail. Consider an input: $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$ where $x_i$is an input of i^{th} image in the dataset $\mathcal{D}$ and comprises a series of $\{x_{i1}, \dots, x_{iT}\}$. Likewise, input$y_i$ is of similar length as x_i and has related labels $\{y_{i1}, \dots, y_{iT}\}$. During FD identification, every input$x_t$denotes temporal information in the image with y_t Being the respective FD class. Figure 1 depicts the block diagram of the proposed FDs identification framework using EPPAL-CNN.

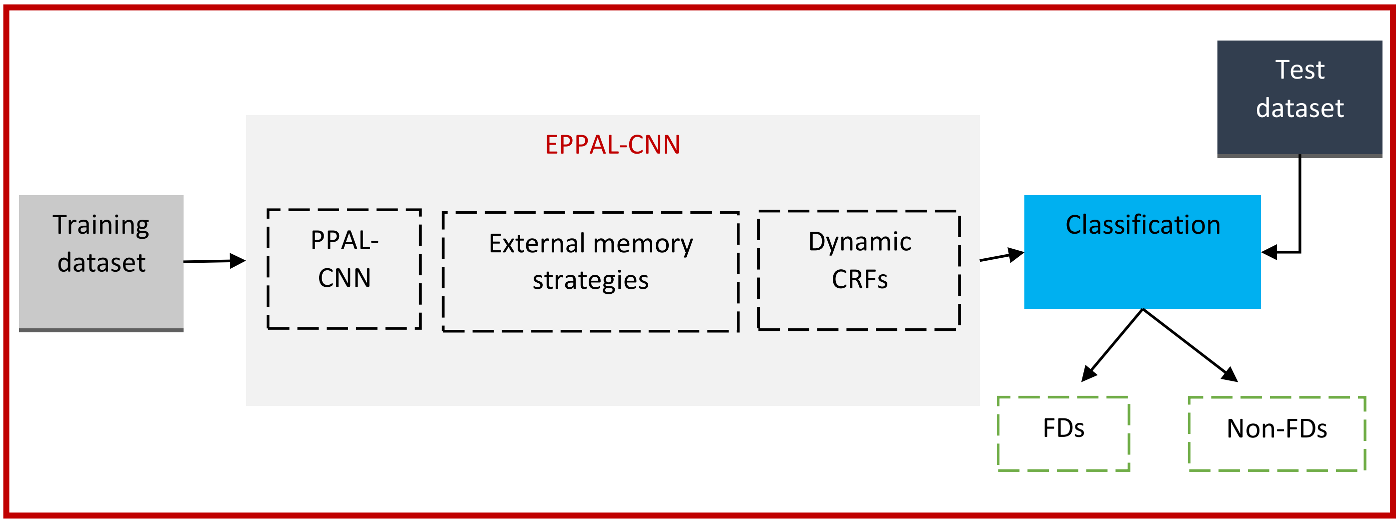

Figure 1. Block Diagram of Proposed FDs Identification Framework

The memory-enhanced DCRF in EPPAL-CNN is portrayed in Figure 2. It is split into memory and DCRF layers. The memory layer is segmented as the input memory $m_{1:t}$, output memory $c_{1:t}$ and a present input u_t i.e., a query in MemNet. The interpretations of input and output memory are associated with attention which determines the weights through relations of input memory and the present input. The outcome of the memory layer is considered as input to the DCRF layer. The below section explains the components of memory-enhanced DCRF in the EPPAL-CNN technique for identifying the FDs.

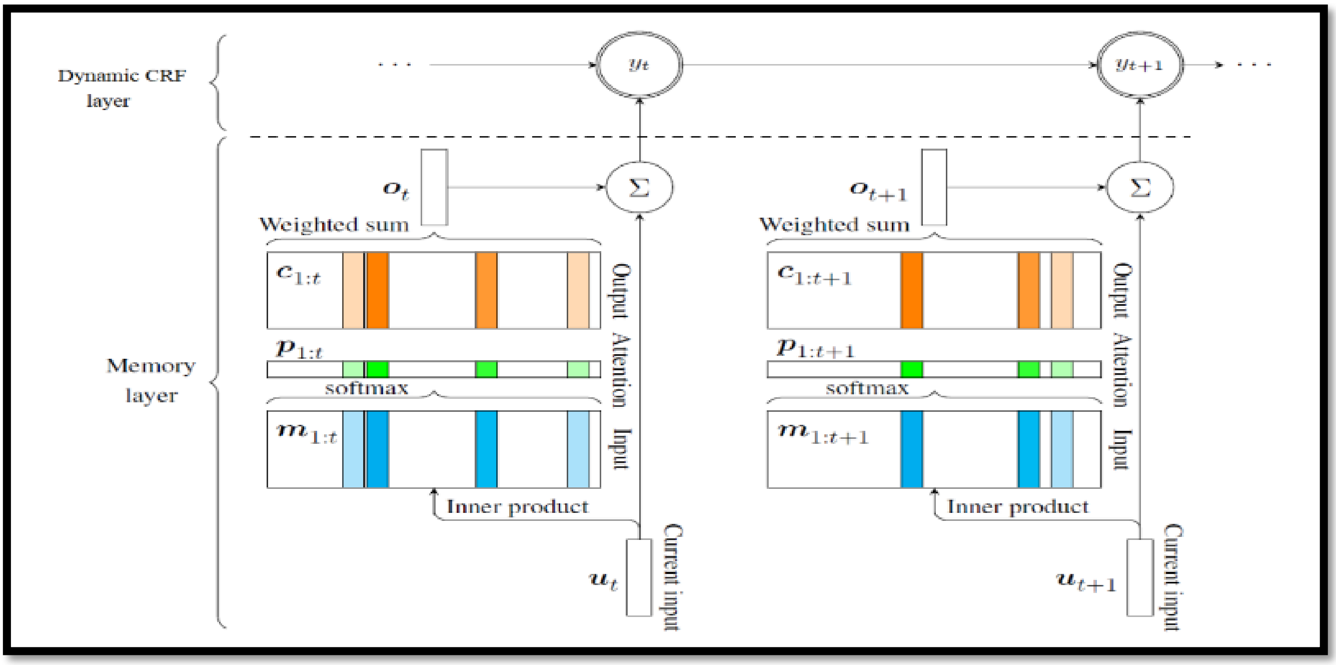

Figure 2. Design of Memory Enhanced DCRFs with a Single Memory Hop in EPPAL-CNN

A. Memory Layer

a) Input Memory

Each pixel in i is encoded with a function $\phi(i_t)$, where $\phi(\cdot)$ maps i_t into a vector that belongs to $\mathbb{R}^d$. The outcome is labeled as$\{i_1, \ldots, i_t\}$. While this enhanced image is observed as a memory in MemNets, the issue becomes insensitivity to the temporal data between memory cells. So, the temporal data is integrated with memory using a bi-directional Gated Recurrent Unit (GRU). The encoding is explained as follows:

$$\vec{m}_t = \overrightarrow{GRU}(i_t, \vec{m}_{t-1}) \quad (1)$$

$$\overleftarrow{m}_t = \overleftarrow{GRU}(i_t, \overleftarrow{m}_{t+1}) \quad (2)$$

$$i_t = \tanh(\vec{W}_m \vec{m}_t + \overleftarrow{W}_m \overleftarrow{m}_t + b_m) \quad (3)$$

In Eq. (3), $\vec{W}_m, \overleftarrow{W}_m$ and b_m Are learnable parameters.

b) Current Input

This is utilized for defining the ongoing stage i_tBe it a defect or defect-free image. Since in MemNets, it is essential to implement the present input to be in a similar region like an input memory; thus, an attention weight of each pixel in the memory is computed via estimating the significance between the two. The current input is defined as $u_t = m_t$.

c) Attention

The significance between the ongoing stage u_t and m_i for $i \in [1, t]$ is determined by SoftMax to measure the attention range of each pixel in the memory.

$$p_{t,i} = softmax(u_t^{\top} m_i) \quad (4)$$

$$\text{Where } softmax(a_i) = \frac{e^{a_i}}{\sum_j e^{a_j}} \quad (5)$$

d) Output Memory

The output memory c_t is calculated analogously, however, using multiple sets of GRUs variables and tanh layers of Eqns. (1)-(3). c_tGives the results of the memory layer and is employed in the DCRF layer as input.

e) Memory Layer Output with Extension

After determining the attention weights, the memory access regulator accepts the response$o$as a weighted sum over the interpretation of output memory:

$$o_t = \textstyle\sum_i p_{t,i} c_i \quad (6)$$

This network model is improved via stacking many memory hops were an outcome of k^{th} hop is termed as input to$(k+1)^{th}$ hop as:

$$u_t^{k+1} = o_t^k + u_t^k \quad (7)$$

In Eq. (7),u_t^{k+1} encodes not only data at the ongoing stage (u_t^k), but also relevant significant information from memory (o_t^k). Here, the number of hops is restricted to 1.

B. Dynamic CRF Layer

After determining the interpretation of u_t^{k+1} and significant data is integrated from memory, and it is fed into the DCRF layer, which generalizes both linear CRFs and highly complicated patterns. EPPAL-CNN considers a factorial CRF consisting of linear label sequences including relations among cotemporally labels. Let a factorial CRF along with$L$sequences wherein$Y_{l,t}$Denotes a parameter in l at interval t. The DCRF's group indexes are $\{(0,l),(1,l)\}$ for every within-sequence margin and $\{(0,l),(0,l+1)\}$For every between-sequence margin.

The factorial CRF G represents a likelihood over hidden states as:

$$\mathcal{P}(y|x) = \frac{1}{\mathcal{Z}(x)}\left(\prod_{t=1}^{T-1}\prod_{l=1}^{L}\phi_l\left(y_{l,t},y_{l,t+1},x,t\right)\right)$$

$$\left(\prod_{t=1}^{T}\prod_{l=1}^{L-1}\psi_l\left(y_{l,t},y_{l+1,t},x,t\right)\right) \quad (8)$$

In Eq. (8), $\{\phi_l\}$and $\{\psi_l\}$ denote the possibilities over within-sequence and between-sequence margins and$\mathcal{Z}(x)$Stands for a split factor. Figure 3 displays a graphical interpretation of factorial CRFs.

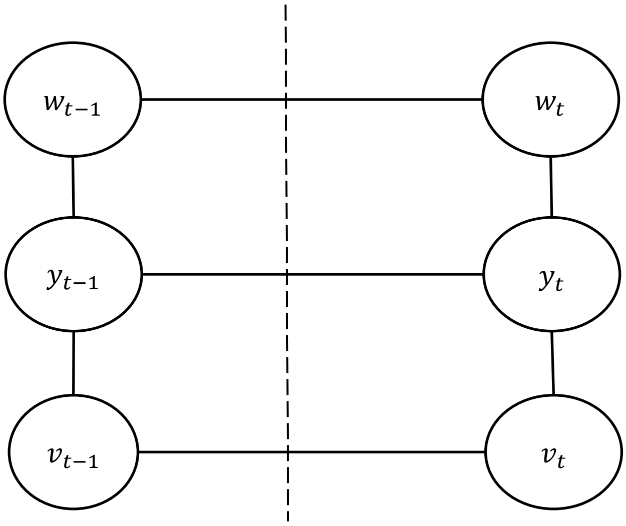

Figure 3. Graphical Interpretation of Factorial CRFs(*the dashed line denotes the edge between timestamps)

The potentials factorize depend on the features $\{f_k\}$ and weights $\{\lambda_k\}$ of G as:

$$\phi_l\left(y_{l,t},y_{l,t+1},x,t\right) = e^{\{\sum_k \lambda_k f_k(y_{l,t},y_{l,t+1},x,t)\}} \quad (9)$$

$$\psi_l\left(y_{l,t},y_{l+1,t},x,t\right) = e^{\{\sum_k \lambda_k f_k(y_{l,t},y_{l+1,t},x,t)\}} \quad (10)$$

Additionally, more complex structures probably rely on how long the sequence is in its present state. This factorized structure utilizes several variables than the state space of the cross-product.

a) Inference in DCRFs

Inference in theDCRF is achieved for unlabeled image$x$to resolve two inference issues: determining the marginal $\mathcal{P}\left(y_{t,c}\middle|x\right)$in each group$y_{t,c}$ and the Viterbi decoding $y^* = \underset{y}{\operatorname{argmax}}\,\mathcal{P}(y|x)$.

It is applied for labeling an unknown image, and marginal determination is applied to estimate the variables. Since marginal determination is required during training, the inference should be effective so that huge training sets are used though if there are several labels. Here, the approximate inference is characterized via Belief Propagation (BP) which iteratively update a vector $i_t = \left(m_u(i_v)\right)$ of temporal data between i_u and i_v. The update from i_u to i_v is as:

$$m_u(i_v) \leftarrow \sum_{i_u}\phi(i_u,i_v)\prod_{i_t \neq i_v} m_t(i_u) \quad (11)$$

In Eq. (11), $\phi(i_u,i_v)$stands for the potential on the margin (i_u,i_v). Executing this modification for single margin (i_u,i_v)separately is known forwarding a piece of pixel information from i_u to i_v. An approximate marginal for i_t is determined as:

$$\mathcal{P}(i_u,i_v) \leftarrow \kappa\phi(i_u,i_v)\prod_{i_t \neq i_v} m_t(i_u)\prod_{i_w \neq i_u} m_w(i_v) \quad (12)$$

In Eq. (12), κ denotes a regularization function. At each iteration of BP, temporal data are forwarded in all ways, and selecting a better plan may influence how quickly the overfitting is reduced. Here, tree-based and random plans are used for BP. The tree-based plan propagates temporal data, including a group of cross-cutting spanning trees of the actual graph. At every iteration of this plan, a spanning tree $\mathcal{T}^{(i)} \in \Upsilon$ is chosen, and temporal data are forwarded in end-to-end with each margin $\mathcal{T}^{(i)}$ to get accurate inference on $\mathcal{T}^{(i)}$. Generally, trees are chosen from any groups$\Upsilon = \{\mathcal{T}\}$ Provided that the trees in Υ comprise the margin group of the actual graph. Practically, trees are chosen randomly, but the initial margins are chosen that were not used in any prior iteration.

The random path easily forwards temporal data across each margin randomly. To enhance convergence, every margin $e_i = (s_i,t_i)$ is randomly ordered, and each temporal data $m_{s_i}(t_i)$is propagated before any data $m_{t_i}(s_i)$. It is also applied to execute Viterbi decoding, but the summation in Eq. (11) is substituted by maximization. As well, inference in DCRFs with higher groups is executed directly by generalized varieties of variational methods.

b) Parameter Determination in DCRFs

The aim of the determination of variables is to obtain a group of variables $\zeta = \{\lambda_k\}$in $\mathcal{D} = \{x_i,y_i\}_{i=1}^{N}$. The conditional log-probability is optimized as:

$$\mathcal{L}(\zeta) = \sum_i \log \mathcal{P}_\zeta(y_i|x_i) \quad (13)$$

The derivative of $\mathcal{L}(\zeta)$regarding λ_k related to group index c is as:

$$\frac{\partial \mathcal{L}}{\partial \lambda_k} = \sum_i \sum_t f_k\left(\vec{y}_{i,t,c}, x_i, t\right) - \sum_i \sum_t \sum_{\vec{y}_{t,c}} \mathcal{P}_\zeta\left(\vec{y}_{t,c}\middle|x_i\right) f_k\left(\vec{y}_{t,c}, x_i, t\right) \quad (14)$$

In Eq. (14), $\vec{y}_{i,t,c}$ denotes the distribution of $y_{t,c}$ in y_i and $\vec{y}_{t,c}$varies over distributions to $y_{t,c}$. Notice that it is the variable $\mathcal{P}_\zeta\left(\vec{y}_{t,c}\middle|x_i\right)$Which desires to determine marginal likelihoods in the unfolded DCRFs.A prior $\mathcal{P}(\zeta)$ is defined over variables and $\log \mathcal{P}(\zeta|\mathcal{D}) = \mathcal{L}(\zeta) + \log \mathcal{P}(\zeta)$ It is adjusted to avoid overfitting. A spherical Gaussian prior is used along average $\mu = 0$ and covariance matrix $\Sigma = \sigma^2 x$, thus a gradient becomes:

$$\frac{\partial \mathcal{P}(\zeta|\mathcal{D})}{\partial \lambda_k} = \frac{\partial \mathcal{L}}{\partial \lambda_k} - \frac{\lambda_k}{\sigma^2} \quad (15)$$

Thus, the CRF in PPAL-CNN is enhanced via combining external memory and DCRFs strategy for identifying the FDs accurately.

Algorithm: EPPAL-CNN for FDs Identification

Input: Training dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$

Output: FDs and non-FDs

```
    Begin
    Initialize $N$ number of training images;
    Apply the EPPAL-CNN;
    //memory enhanced DCRFs in EPPAL-CNN
    Initialize the learnable parameters;
    Integrate the temporal information with memory;
    Determine the input memory via Eqns. (1)-(3);
    Denote the current input as $u_t = m_t$;
    Measure the significance between input memory &current input via Eq. (4);
    Obtain the output memory via Eqns. (6)-(7);
    Apply the output memory as input to the dynamic (factorial) CRFs;
    Determine the distribution over hidden states as Eq. (8);
    Compute the potentials factorize using $\{f_k\}$ and $\{\lambda_k\}$ as Eqns. (9)-(10);
    Update $i_t$ of temporal data between $i_u$ and $i_v$ as Eq. (11);
    Calculate the approximate marginal for $i_t$ using Eq. (12);
    Optimize the conditional log-likelihood via Eqns. (13)-(15);
    Execute the EPPAL-CNN;
    Identify the FDs and non-FDs;
    End
```

IV. RESULTS& DISCUSSIONS

This section presents EPPAL-CNN performance via implementing it in MATLAB 2017b with the help of TILDAin, in which every image comprises a text reporting flaw regions in it. TILDA includes an overall of 8 representative textile categories and also 8 sorts of classes for every category of textile that has existed. So, an entire database has 3200 TIF images having a total capacity of 1.2GB. Also, its efficiency is analyzed with PPAL-CNN based on precision, recall, f-measure, and accuracy. The confusion matrix for every class is obtained independently, and an average of identified outcomes for EPPAL-CNN is depicted in Table 1.

Table 1. Confusion Matrix for 1100 Test Images

	Identified Class		
		Positive	Negative
Actual Class	Positive (547 for each class)	True Positive **512**	False Negative **35**
	Negative(550 for other class)	False Positive **38**	True Negative **512**

A. Accuracy

It is the percentage of recognizing FDs accurately over the sum amount of attempts executed.

$$Acc = \frac{True\ Positive + True\ Negative}{TP + TN + False\ Positive + False\ Negative}$$

TP gives a result where the EPPAL-CNN properly identifies FDlabels as FDs. FP gives a result where the EPPAL-CNN inexactly detects FDlabels as non-FDs.FN gives a result where the EPPAL-CNN inexactly detects non-FDlabels as FDs.TN gives a result where the EPPAL-CNN exactly detects non-FDlabels as non-FDs.

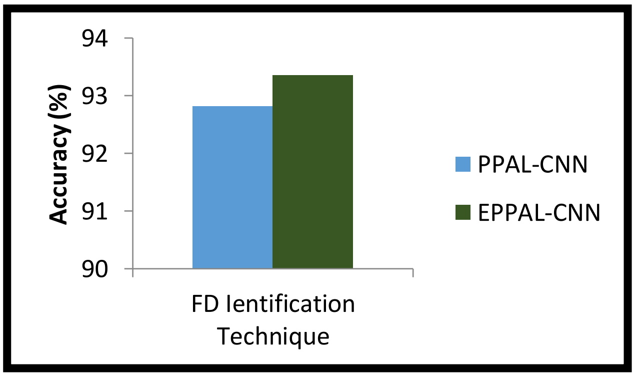

Figure 4. Comparison of Accuracy

Figure 4 portrays $Acc(\%)$ of EPPAL-CNN and PPAL-CNN techniques. It indicates that$Acc$of EPPAL-CNN is 0.59% higher than PPAL-CNN because of applying DCRF and external memory strategies.

B. Precision

It is some exactly identified FD classes at TP and FP.

$$Prsn = \frac{No.of\ exactly\ identified\ FD\ classes}{No.of\ exactly\ identified\ FD\ classes + No.of\ inexactly\ identified\ FD\ classes}$$

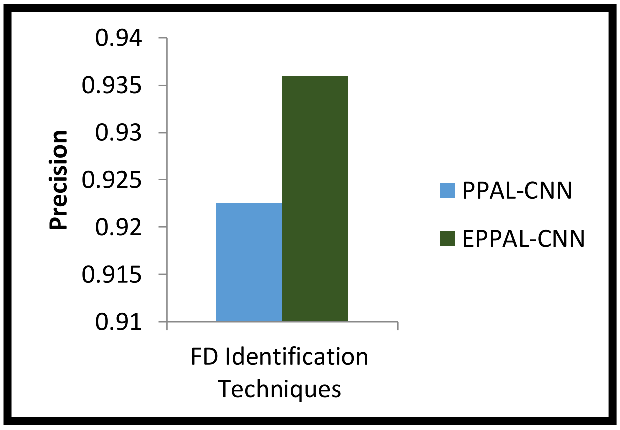

Figure 5. Comparison of Precision

In Figure 5, $Prsn$of EPPAL-CNN and PPAL-CNN techniques are depicted. It notices that $Prsn$of EPPAL-CNN is 1.46% greater compared to PPAL-CNN owing to the consideration of enhancing CRF in PPAL-CNN.

c) Recall

It is the rate of correctly identifying the FD classes at TP and FN.

$$Recall = \frac{No.of\ exactly\ identified\ FD\ classes}{No.of\ exactly\ identified\ FD\ classes + No.of\ inexactly\ identified\ non{-}FD\ classes}$$

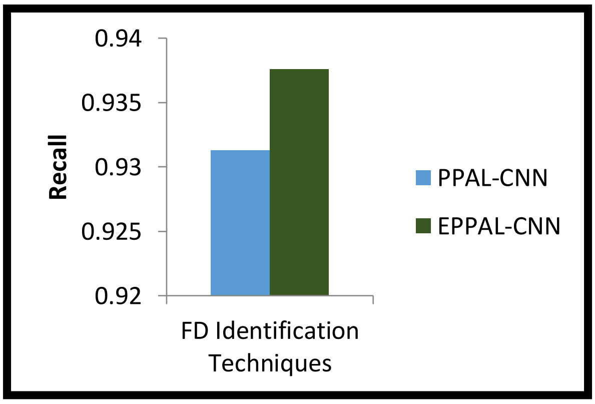

Figure 6. Comparison of Recall

Figure 6 portrays the recall of PPAL-CNN and EPPAL-CNN techniques. It denotes that the recall of EPPAL-CNN is 0.68% maximized than PPAL-CNN. The reason is handling the complex structure interaction of FDs using DCRF, including external memory strategies efficiently.

D. F-measure

It defines a harmonic average of precision and recall.

$$F - measure = 2 \times \frac{Precision \cdot Recall}{Precision + Recall}$$

In Figure 7, the f-measure of PPAL-CNN and EPPAL-CNN techniques are depicted. It concludes that the f-measure of EPPAL-CNN is 0.56% increased than the PPAL-CNN.

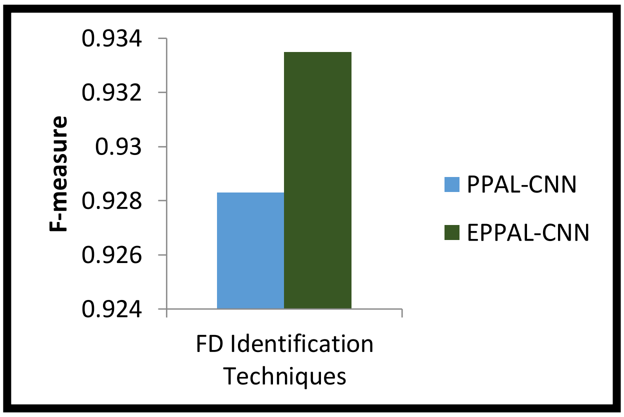

Figure 7. Comparison of F-measure

V. CONCLUSION

This article suggests the EPPAL-CNN method to control the complex structure interaction of FDs. At first, the CRF is developed via combining external memory strategies encouraged from the memory networks and so allowing the CRFs for representing the localized features and using a complete image. It comprises memory and DCRF layers. The memory layer is partitioned into input, output, and present input memory. The interpretations of input and output memory are associated through attention in which weights are calculated by the relevance of input and present input memory. Then, the DCRF layer is implemented by taking the outcome of the memory layer as its input defines the distributed hidden state and complex relation between labels. Its factorial construction includes the relations among cotemporally labels, explicitly modeling constrained likelihood dependencies among various labels. Using EPPAL-CNN, a higher-order Markov dependence between labels is modeled by taking into consideration of the external memory. To end, the investigational outcomes proved that the EPPAL-CNN had achieved 93.36% accuracy, which is 0.6% larger compared to the PPAL-CNN technique for the TILDA database. But it causes undesirable convergence behavior and unwanted estimations. So, future research will solve these problems using weight optimization methods.

ACKNOWLEDGMENT

The authors would like to thank KarpagamUniversity, Coimbatore, Tamil Nadu, India, for supporting the research and provide facilities.

REFERENCES

[1] Ngan, H. Y., Pang, G. K., & Yung, N. H. (2011). Automated fabric defect detection—a review. Image and Vision Computing, 29(7) 442-458.

[2] Tsang, C. S., Ngan, H. Y., & Pang, G. K.. Fabric inspection based on the Elo rating method. Pattern Recognition, 51 (2016) 378-394.

[3] Long, J., Shelhamer, E., & Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (2015) 3431-3440.

[4] Ronneberger, O., Fischer, P., &Brox, T. U-net: convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, Cham, (2015) 234-241.

[5] Badrinarayanan, V., Kendall, A., & Cipolla, R. SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12) (2017) 2481-2495.

[6] Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., ... & Torr, P. H. Conditional random fields as recurrent neural networks. In Proceedings of the IEEE International Conference on Computer Vision, (2015) 1529-1537.

[7] Çelik, H., Dülger, L. C., Topalbekiroglu, M., & Rosa, J. L. G. Application of neural networks (NNs) for fabric defect classification. In Artificial Neural Networks Models and Applications. InTech(2016).

[8] Ouyang, W., Xu, B., Hou, J., & Yuan, X. Fabric defect detection using activation layer embedded convolutional neural network. IEEE Access, 7(2019)70130-70140.

[9] Liu, Z., Li, C., Zhao, Q., Liao, L., & Dong, Y. A fabric defect detection algorithm via context-based local texture saliency analysis. International Journal of Clothing Science and Technology, 27(5) (2015) 738-750.

[10] Li, Y., Zhao, W., & Pan, J. Deformable patterned fabric defect detection with fisher criterion-based deep learning. IEEE Transactions on Automation Science and Engineering, 14(2) (2017) 1256-1264.

[11] Jia, L., Zhang, J., Chen, S., & Hou, Z. Fabric defect inspection based on lattice segmentation and lattice templates. Journal of the Franklin Institute, 355(15) (2018) 7764-7798.

[12] Yapi, D., Allili, M. S., &Baaziz, N. Automatic fabric defect detection using learning-based local textural distributions in the contourlet domain. IEEE Transactions on Automation Science and Engineering, 15(3) (2018) 1014-1026.

[13] Mei, S., Wang, Y., & Wen, G. Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model. Sensors, 18(4) (2018) 1064.

[14] Jeyaraj, P. R., & Samuel Nadar, E. R. Computer vision for automatic detection and classification of fabric defects employing deep learning algorithm. International Journal of Clothing Science and Technology, 31(4) (2019) 510-521.

[15] Lin, S., He, Z., & Sun, L. Defect enhancement generative adversarial network for enlarging data set of microcrack defect. IEEE Access, 7, (2019) 148413-148423.

[16] Hu, G., Huang, J., Wang, Q., Li, J., Xu, Z., & Huang, X. Unsupervised fabric defect detection based on a deep convolutional generative adversarial network. Textile Research Journal, 90(3-4) (2020) 247-270.

[17] Jing, J., Wang, Z., Rätsch, M., & Zhang, H. Mobile-Unet: an efficient convolutional neural network for fabric defect detection. Textile Research Journal, (2020) 1-13.