Image Analysis using Machine Learning and Traditional Image Processing Approaches

Puneeth Kumar B S*1, Mr S.V.Ramanan2, Galiveeti Poornima3, Mrs. Jayashree M Kudari4, Dr. C M Velu5, Hari Krishna silamanthula6

*1Assistant Professor, Department of Computer Science, St. Joseph's University, Langford road, Bengaluru, Karnataka, India.

2Assistant Professor, Department of Electronics and Communication Engineering, PPG Institute of Technology, Coimbatore, Tamil Nadu, India.

3Assistant Professor, Department of Computer Science and Engineering, Presidency University, Bangalore, Karnataka, India.

4Associate Professor, School of CS and IT, Jain Deemed to be university, Bangalore, Karnataka, India.

5Professor, Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha University, SIMATS, Chennai, Tamilnadu , India.

6Assistant Professor, Department of Fine Arts, K L E F Deemed to be University, Guntur, Andhra Pradesh, India.

Abstract

The demand for speedier algorithms that can extract information from photos is becoming more crucial due to the continuously growing amount of image data in practically all disciplines. The practise of utilising digital image processing methods to extract significant information from a 2-dimensional picture is known as image analysis. The term "digital image processing" refers to a group of methods used in pre-processing (such as noise removal and image enhancement), image compression, feature extraction (such as edges and contours), and various key point detection methods (such as corners and joints, areas with particular colours or textures). This study focuses on the recognition and representation of texture, a crucial aspect of picture analysis. Utilizing a well-known texture identification technique built on filter banks, we conduct tests on a variety of artificial and real-world photos. We pinpoint certain particular situations in which the algorithm fails and suggest a change to the original approach that produces better segmentations.

Keywords: *Machine Learning, Digital Image, Filters, Feature extraction and Noise .*

DOI Number: 10.14704/nq.2022.20.9.NQ44537 **Neuro Quantology 2022; 20(9):4638-4650**

1. Introduction

The practise of utilising digital image processing methods to extract significant information from a 2-dimensional picture is known as image analysis. The term "digital image processing" refers to a group of methods used for preprocessing (such as noise removal and image enhancement), image compression, feature extraction (such as edges and contours), and various key point detection methods (such as corners and joints, areas with particular colours or textures) [1]. The processes involved in image analysis now include the following thanks to the development of machine learning (ML) and deep learning (DL) methods, which are the major topics of this paper: Choose a machine learning (ML) algorithm that is appropriate for the task at hand, such as Support Vector Machine (SVM), Naive Bayes (NB), Expectation Maximization (EM), or k-means, or a suitable deep learning architecture, such as Convolutional Neural Network (CNN), Long Short Term Memory (LSTM), or Autoencoder. The task at hand, the training data that are available, and the computational resources that are available all play a role in selecting the DL architecture or ML method [2]. Preprocessing and feature engineering are two terms that refer to the preparation of training data for the ML/DL model. In the case of DL models, there is very no pre-processing, such as data whitening, and there is no feature engineering module since the model learns the features. Extract relevant characteristics from the picture in the case of ML [3]. The Histogram of Oriented Gradients (HOG), Scale Invariant Feature Transform (SIFT), Fisher vector, and deep convolutional features are a few examples of frequently utilised features.

Create a model using the training data, then test it using a different validation dataset [4]. Apply post processing techniques, such as thresholding, to convert the model predictions into the required output format. Figures 1. and 2., respectively, depict a typical ML and DL pipeline for a general image analysis problem.

Figure.1. Pipeline typical for machine learning. The post-processing module then carries out activities like thresholding after the Feature Engineering module has extracted features from the raw data and the ML model has predicted the results.

The function of conventional image processing is reduced to the following use cases as ML and DL progress and take over the work of picture analysis: 1. Pre- and post-processing for feature engineering (for ML). 2. Analyzing the learnt hidden layer features in the case of DL in order to use the intermediate features to explain the prediction. 3. Being aware of the training dynamics, or how the data's topology evolves throughout the course of training epochs.

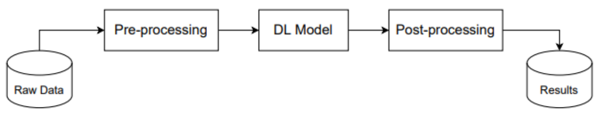


Figure.2. An example of a deep learning pipeline. The DL pipeline is different from the ML pipeline in that the early layers of the DL model learn the features instead of having a Feature Engineering module.

Image Classification

When compared to all the other activities required in interpreting a digital picture, image classification is the easiest [5]. The process of giving a picture one of k possible labels is known as image categorization. It is a mapping from an image to items in a discrete set with the class labels "c1, c2, c3,..., ck" and k distinct class labels. A multi-class classifier is what this kind of classifier is often referred to as, while a binary classifier is what it is when k = 2. As illustrated in Figure.3, a multi-class classifier is often constructed as a module that generates k distinct probabilities, p(ci) 1 ≤ i ≤ k.

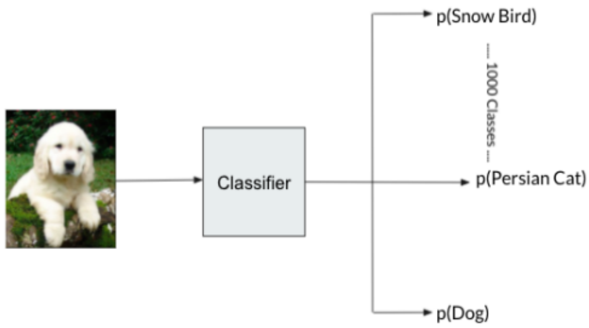


Figure.3. Image Classification. In multi-class image classification an image is assigned one of k (k = 1000 in this case) different labels.

Because the output classes in multi-class classification are mutually exclusive, the total probability of all outputs is 1. Typical applications of picture categorization include the following: 1. Sort a natural picture into a category depending on the thing it contains [6]. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC), which was held annually from 2010 to 2017, using a portion of the ImageNet dataset, is a well-known example of a large-scale image classification challenge (classifying 1 million pictures into one of 1000 class labels). The ImageNet is a sizable library of photographs from many categories that are arranged hierarchically in accordance with the WordNet structure [7]. Each node in the hierarchy is represented by hundreds to thousands of photographs in the database. At the moment, the WordNet only contains nouns. 2. Determine if a medical picture of a person obtained via magnetic resonance imaging (MRI) or computed tomography (CT) scanning is healthy or sick. 3. Determine if a medical picture is diagonalizable or not. 4. Identify the kind of document image according to the document type (e.g., Research Articles, Legal documents and Resume).

Object Detection in images

Finding a specific item in a picture is known as object detection. The objective is often to locate the area in a picture where an instance of an item is visible [8]. The job of the detection method is to locate the item in the picture (typically as a bounding box, as illustrated in Figure.4), while image classification techniques verify whether an object is there or not [9]. Optical character recognition (OCR), face identification, and the detection of diverse structures like tables, figures, and charts in document pictures are all examples of applications for object detection.

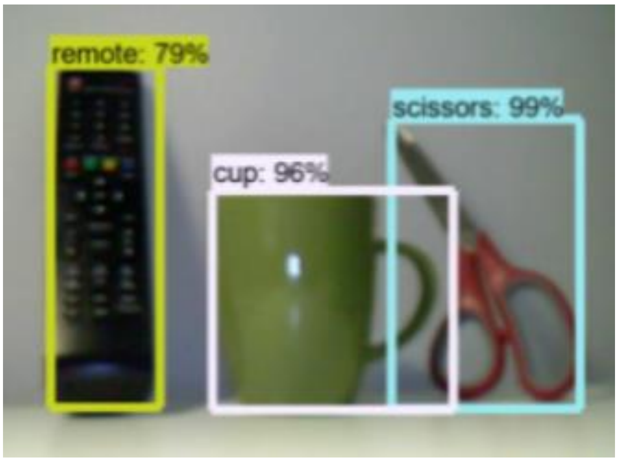


Figure.4. Object recognition The process of identifying a bounding box and a confidence level for each item in the picture is known as object detection.

Image Segmentation

The next degree of complexity after image classification and detection is to segment the picture into k separate segments, or to provide a label of l, l c1, c2, c3,..., ck, to each pixel in the image [10]. While classification and object recognition may be performed on a lower resolution version of the picture (to speed up calculation), image segmentation must typically be performed at the original resolution or a resolution that is very near to it [11]. The following categories may be used to categorise the picture segmentation task according to the amount of information they convey: Semantic segmentation, first 2. Segmenting by instances Panoptic segmentation, third 4. Segmentation of parts.

The original picture (Figure.5a) is segmented into several areas depending on the kind of item (person) as shown in Figure.5b. This process is known as semantic segmentation, also referred to as class level segmentation [12]. Each instance of a class person is segmented as a distinct segment under instance segmentation, as illustrated in Figure.5c. As shown in Figure.5d, panoptic segmentation combines semantic and instance segmentation by predicting the identities of both instances and objects. Each component of an instance of an object is split into a distinct segment in part segmentation, as shown in Figure 6.

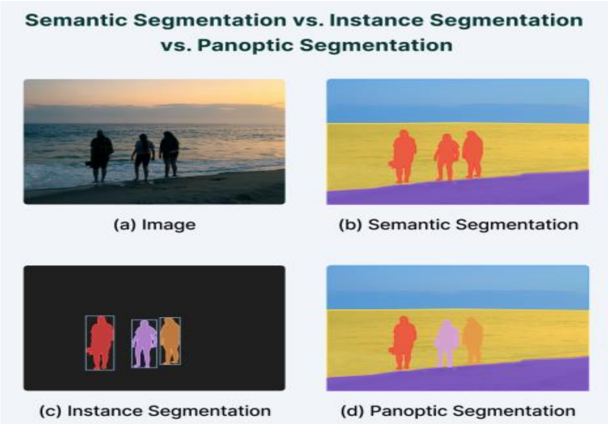


Figure.5. Depending on what is segmented, several types of picture segmentation are used. While each individual is given a separate label in instance segmentation, all instances of people in the picture are given the same label in semantic segmentation. Semantic and instance segmentation are combined in panoptic segmentation.

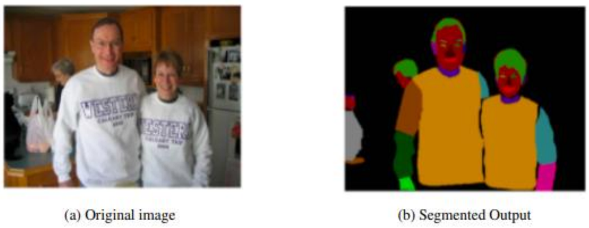


Figure.6. Segmentation in part. Different areas of an item are segmented in part segmentation.

In computer vision and medical image analysis, picture segmentation is sometimes used for the following purposes: 1. In medical pictures, segmenting an organ or anomalies like a tumour area. 2. Image segmentation is a crucial part of robotics because it allows robots to draw boundaries between different areas [13]. 3. Image segmentation is a technique used in satellite image analysis to distinguish between different areas, such as water bodies, forests, highways, and buildings.

Methodologies and challenges in Image Analysis

A taxonomy of numerous image processing and analysis approaches is shown in Figure 8. Traditionally, a variety of digital image processing approaches have been used to tackle picture classification, detection, and segmentation challenges [14]. However, before 2012, hand-crafted features like HOG, SIFT, and key-points, combined with machine learning (ML) algorithms like k-means, Gaussian Mixture Model (GMM), or SVM were used. Numerous supervised and unsupervised DL models have replaced unsupervised DL models as a result of AlexNext's achievement in decisively defeating the ILSVRC challenge utilising deep neural networks that can learn both the task's requirements and its features at the same time.

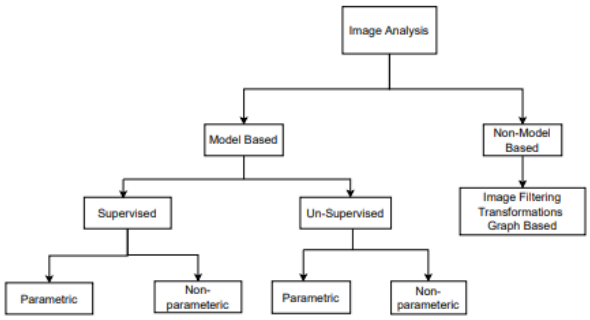


Figure.8. Different types of image analysis techniques

methods not dependent on models These are algorithmic techniques that start with an input picture and then perform a series of specified alterations on it to get the desired result [15]. Clustering pixel-level characteristics and graph-based methods like normalised cuts are a few of the common examples. The algorithmic approach's primary flaw is that it can't effectively generalise to various kinds of photos. It is crucial to convert the input image's representation from its native pixel domain to a feature domain representation in order to generalise these techniques [16]. There are local and global features. Small areas may use local characteristics (often known as super pixels). Edges, corners, joints, and textures are a few examples of local characteristics. Key points are a generalisation of other kinds of characteristics, such as joints and corners. Both the full picture and considerably bigger parts of the image may benefit from global characteristics [17]. Examples include different colour or intensity statistics, outlines, and forms. However, unlike the model-based techniques outlined in the next section, these approaches do not need training photos [18]. Examples of segmented pictures employing texture and contour characteristics are shown in Figures 9a and 9b, respectively. The fact that different photos pair well with various attributes is one of the difficulties with feature-based techniques, as is seen from the figure. Finding a collection of characteristics that will work for all of the photographs being considered is a difficult undertaking. Deep learning models have replaced the conventional feature-based techniques as a result of this and the effectiveness of deep learning in automatically learning the features from a collection of training data.

(a) Segmentation using texture (b) Segmentation using contour

Figure.9. Examples of photos that may be segmented using texture and contour clues. While Figure (b) displays a sample picture where segmentation using a contour-based approach would perform better, Figure (a) offers an example of an image where segmentation using a texture cue would perform better.

Model-based methods Model-based methods use a series of training photos to develop a statistical or mathematical model, while non-model techniques use a set of image modification algorithms and graph-based optimization techniques (min-cut, max-flow) [19]. A model is a mathematical function that takes an input vector, x, and computes an output vector, y, using the formula y = f(x). Training is the process of utilising examples from the training set to identify the best model that minimises the empirical loss and generalises well [20]. Any sample input picture may be used to predict the output after a model has been trained and refined.

The method for completing image analysis jobs has undergone a constant paradigm change throughout the last 20 years [21]. The signal processing-based techniques (such as image transformations) and graph-based methods that were popular in the 1990s were replaced in the early 2000s by machine learning-based approaches using different hand-crafted features (such as HOG, SIFT, and SURF) [22]. The Viola Jones face detection method and the Deformable Parts Model (DPM) for person identification are two examples of such successful models [23]. However, the effectiveness of the characteristics is what determines how well these techniques function, thus the study was concentrated on finding features that were more effective. Starting in 2012, there was a significant improvement in computer vision task performance. A deep (12 layer) convolutional neural network (CNN) with 1000 discrete classes was trained by Alex Krizhevsky and his colleagues to classify photos from ImageNet. Their model, which was trained on 1 million unprocessed photos from ImageNet, won the 2012 ILSVRC competition with an error rate that dropped significantly from 25.8% (in 2011) to 16.4%. Since then, the most promising method for a variety of image analysis tasks has been deep learning models, which are data-driven, learn the task-specific features from raw data and execute the job using the learned features [24]. Given that this framework is used to tackle complicated issues and that the model learns features from raw data, deep learning models display the following traits: 1. Because the models contain so many parameters, they need a tonne of training data. 2. To get the best results, DL models must be trained for a lengthy time (from days to weeks) on equipment with a lot of memory and quick processing power. Typically, hardware accelerators like Tensor and Graphics Processing Unit (GPU) The majority of machine learning models also learn in a supervised fashion, requiring for training both the input pictures and the predicted output for each image. Even while unsupervised DL models like VAE and GAN are common, classification, object identification, and segmentation tasks do not benefit from the information that these models employ.

2. Literature Survey

For image analysis, features calculated from the raw input picture give a wealth of extra information that is not explicitly stated in the original representation. Global features are those calculated from the whole picture, while local features are those calculated from only a tiny portion of each pixel's N(x, y) neighbourhood [25]. Edges, focal points, various colour spaces, roughness, curves, and so forth are a few examples of local characteristics. Transposition and Convolution The most common image modifications used to extract local information from a picture are convolutions [26]. Intensity and colour statistics, Scale Invariant Feature Transform (SIFT), and other pooling methods like Vector of Locally Aggregated Descriptors (VLAD) and Fisher Vector are examples of global features [27]. To create a VLAD representation of a picture, d-dimensional local features must first be calculated (like for example filter bank response at each pixel) [28]. By applying k-means clustering of local characteristics and then calculating k d-dimensional vectors, which are the sum of the distance between each sample and the associated nearby code-word, a collection of k d-dimensional codewords are created (cluster center). The final k d-dimensional vectors are properly normalised and utilised to represent the picture in a different way. Similar to how a Fisher Vector (FV) is created, a normalised gradient of log likelihood with respect to parameters is computed after first using GMM to model the distribution of local features (mean, variance and mixture coefficients). Medical image analysis tasks are different from tasks linked to analysis of non-medical pictures analysed using a standard Red Green Blue (RGB) camera in the following ways: 1. As contrast to qualitative outcomes in non-medical pictures, quantitative output is a crucial prerequisite for medical image analysis. For instance, evaluating the severity of an illness and calculating numerous morphometric characteristics, among other things. 2. A lot of medical imaging techniques, including MR and CT scans, work with multiple perspectives and 3-D volumes [29]. In contrast to non-medical scenarios where segmentation of 2-D pictures is anticipated, the segmentation or detection operation is intended to provide a 3-D volume as the result. 3. Compared to medical pictures, whose support of distribution is significantly smaller than real-word images, the distribution of non-medical images has a far wider variation. 4. While there is a wealth of image data available for non-medical image analysis and getting the data annotated is easier by methods like crowdsourcing as no specific domain knowledge is required, it is challenging to obtain training data for medical images due to ethical issues and further creating the ground truth annotation, as required by supervised models. 5. Depending on the imaging modality, specific correction approaches are required for artefacts caused by the picture capture equipment. Medical and non-medical image analysis have historically been two distinct fields because of the aforementioned variances [30]. Using Deep Neural Networks for segmentation tasks presents a number of difficulties, including the loss of finer features like edges due to the decline in featuremap resolution as one goes deeper. The U-Net model, which was created especially for medical picture segmentation, features many skip connections that transport fine-grained information from beginning layers to subsequent layers in order to get around this. Joshua recently demonstrated state-of-the-art accuracy when segmenting blood vessels in retinal pictures using a modified version of U-Net. A series of several U-Nets are cascaded together in Zhuang's Laddernet invention. A multi-kernel pooling block is used in M-GAN, developed by Kyeong et al., to allow scaleinvariance, a property that is greatly desired for blood vessel segmentation. Sun developed a brand-new data augmentation method that may be used to the segmentation of blood vessels and published findings using the DRIVE, CHASE DB, and STARE datasets. Magnetic Resonance Imaging, which is the most often used non-invasive method for detecting cancer and many other ailments, is another well-liked medical imaging modality. The capacity of MR imaging to provide high contrast pictures, particularly for structures composed of soft tissues like the Corpus Callosum, Gray Matter, White Matter, etc. in the brain, is what gives it its significance. As a result, it is often used to identify illnesses of the spine and brain. An MRI scanner produces three-dimensional (3-D) pictures of internal organs along a variety of orthogonal orientations, with sagittal, coronal, and axial orientations being the most common. The various slices of 2-D photos used to create 3-D images along each of these viewpoints. 2-D The 3-D volumetric data provided in MRI is not used very efficiently by CNN. In particular, 2-D convolutional layer features are independent of their third-direction third-order neighbours. Deep Neural networks with 3-D convolution were invented to get around issue. Mehdi et al. segment 4 distinct subregions of tumour necrotic and the non-enhancing tumour core, peritumoral edoema, and enhancing tumour using two parallel 3-D convolutional networks, one to capture local characteristics and the other to collect the global features of the whole volume. Xiangyu and others

For tasks like object identification and picture segmentation, texture is a crucial indication. Contrary to other signals like colour, contour, and form, texture is far more varied in real-world photographs, making it difficult to identify and characterise. Material identification, picture inpainting, and quality checking are a few uses for texture detection. According to Mircea Cimpoi et al., there are two alternative ways to describe texture. They provide texture classification results utilising this representation on a variety of material detection datasets. 1) Improved Fisher Vector (IFV) pooling of dense-SIFT and colour features. 2) Features derived from a pre-trained convolutional neural network. Bilinear CNN models are introduced by Lin et al. in their work. These models are made up of two feature extractor networks, whose outputs are merged using outer product at each spatial point. As an extension of methods like Fisher Vector and VLAD, they offer it. Based on Fisher Vector and Bilinear CNN models, Lin et al. examine the translational invariance of different texture representations in their work. They also provide a method for inverting a texture representation and seeing the matching pre-image in the same work. In this study, Chenyun et al. offer a method for assembling various qualities to describe textures and for creating synthetic textures based on such descriptions. They provide a brand-new dataset called Describable Textures in Detail Dataset in this paper. For the tasks of table identification and table structure detection, a number of Deep Neural Network topologies have been investigated. Region with Convolutional Neural Network is the first Deep Learning architecture to be used for an object identification problem (R-CNN). Sebastian et al. deploy a different fully convolutional neural network (FCNN), FC-X, for segmenting rows, columns, and cells in addition to Faster-RCNN, a faster version of R-CNN, for table identification. White spaces in the document picture, as opposed to the content, were employed by Raskovic et al. as heuristics for identifying borderless tabular data. With a base network as a VGG and two distinct branches at the end—one for table identification and the other for table column recognition—Paliwal et al. trained a single multi-tasking Deep Learning model. In their most recent research, Xu et al. present the PubTabNet dataset, the Encoder Dual Decoder (EDD) model, and the tree-edit-distance evaluation measure for the job of detecting table structures. An encoder, a structure decoder, and a cell decoder make up their model. In a different recent study, Cascade mask R-CNN, a multi-stage CNN model using HRNet as its backbone, was trained iteratively for both table and structure identification. The model is first trained using a generic dataset, and it is then repeatedly fine-tuned for particular tasks using an iterative training method they call two-stage transfer learning. By integrating the signals gained from both top-down (cell identification) and bottom-up analyses, Sachin Raja et al. are able to solve the difficulty of detecting table structure (structure detection). They use a modified form of Masked-RCNN for the top-down cell identification issue. To identify row and column structure, they use Graph Neural Networks for structure identification. Together, they train the two networks. For accomplishing tasks like classification, object identification, and segmentation in photos and videos, deep neural networks have been effectively deployed. Variational Autoencoder (VAE) and Generative Adversarial Networks (GAN) are the most popular examples of multi-layer Deep Neural networks that have recently been shown to be good options for generative models. All of these models are founded on the fundamental idea that it is possible to learn a complicated, parameterized function through back propagation. These networks are capable of learning functions that generalise effectively when trained with a large quantity of data over a sufficient number of training runs and with the right regularisation methods.

To get around the problem of a lack of labelled data, a number of alternative strategies have been investigated, including semi-supervised learning, Active Learning (AL), self-supervised learning, and few-shot learning. In semi-supervised learning, there is a mix of labelled and unlabeled data in the training set, and the model attempts to extract as much information as it can from the unlabeled data as well. An interactive and iterative method called active learning selects certain samples from the unlabeled pool for expert annotation after each iteration. The ultimate objective is to annotate the fewest number of samples necessary to get the highest performance. The model creates the label from the unlabeled data during self-supervision. One example is to change the data according to a known algorithm and then use the kind of transformation as the label for supervised training.

3. Texture Analysis on Real World Images

Prior to the advent of deep learning, heuristic (or hand-crafted) features and much simpler ML models, such as SVM (for classification), K-means (for segmentation), etc., were used for all image analysis tasks. These methods begin the analysis of pictures by locating numerous low-level elements including edges, contours, and textures. Further high level tasks like categorization, object identification, and segmentation need the use of such characteristics. Even with deep learning, which involves the training of a deep neural network and the learning of a set of hierarchical features, the analysis of the learnt features aids in the explanation of the predictions. Heuristic characteristics include, for instance, colour values (in several colour systems), intensity levels, textures, and edges. Because of its richness, texture is one of the most critical of them (texture is a crucial visual clue for humans while looking at images). But since the texture is so hard to define, it is also very difficult to extract the texture. Images may be transformed in a variety of ways to emphasise different features of the picture. For instance, applying certain filters to a picture, such as the Sobel filter, will result in a filtered image in which the value at a certain spatial point clearly indicates the presence of an edge at that place (whereas in the original image domain representation, it is not possible to identify whether there is an edge or not just by looking at the pixel value at one single location). This section looks at a few methods for making the existence of various textures clear. We also take a closer look at a few uses of texture-based categorization for MR images. Although texture is a crucial element that may be utilised for a variety of image analysis tasks, it can be difficult to identify and depict due to the following difficulties: 1. It is impossible to provide a precise explanation of what a "textured area" is. 2. Natural and other photos have a wide range of textures, each with its own size and pattern. It is so difficult to create an algorithm or filter mask for texture identification and representation. Below, in Figure.10, are several samples of textures. The Figure clearly shows how textures differ greatly in size and kind (stochastic vs deterministic etc.).

Figure.10. Examples of different types of textures

4. Texture Detection and Representation

Similar to edges, it is impossible to determine the kind of texture by examining the value at a single spatial position. In order to determine the texture, it is required to look at the neighborhood's pattern. Contrary to edges, one of the major difficulties in texture detection is the extensive diversity in the texture. Due to the fact that no one filter is capable of identifying all sorts of textures, it is hard to construct certain filters (such the Sobel filter for edge detection) in advance for texture detection. The search for a stable set of basis filters that can recognise the widest variety of textures, however, has been ongoing. Utilizing a certain number of edge detectors (gradient filters) with varying size and orientation is one way to do this. The representational capability of this method is shown by the texture detection technique discussed in this section. One might draw the conclusion that it is impossible to create a single texture detection filter after considering the potential differences in texture type. The issue of whether there is a family of parameterized filters that can be used to identify as many textures as are present in real-world use cases naturally arises. It makes sense to assume that a group of filters (referred to as a "filter bank" from here on) that can capture gradients of pixel intensities at different scales and orientations can be used to detect and characterise the textures that occur in images of practical interest. This is because one way to characterise texture is by looking at the intensity gradient. One of the first efforts in this field was by Jitendra Malik et al., who suggested a method to locate and describe textured regions in a picture by grouping responses of filtering the image by a bank of oriented gaussian filters at various scales and orientations. We investigated this technique further and came up with a variant that calls for absolute value of the filter response to be taken prior to clustering. As the absolute value function is a nonlinear function, our adjustment is a prime illustration of how the representational capacity may be increased by adding nonlinear filters (which is the fundamental idea behind how multi-layer neural networks operate).

We provide further information on how to calculate the hyper parameters, such as the scale range and the number of orientations in the filter bank, in addition to offering a change to the original technique. The two most important steps in automating these parameters are 1) Identifying their impact and 2) Developing an estimation strategy for the ideal parameter values. The method presented in [15] has the following parameters: 1. Number of orientations 2. Value range to be applied as a scale 3. Size of a scale-dependent filter mask 4. Oriented Gaussian derivative filter elongation ratio (x/y) 5. The number of octaves that separate subsequent scale parameters.

5. Data Description

In order to find the best texture detection method and parameter tweaking, we ran texture detection tests using datasets created artificially. The final algorithm that produces the best results has also been tested on various real photos.

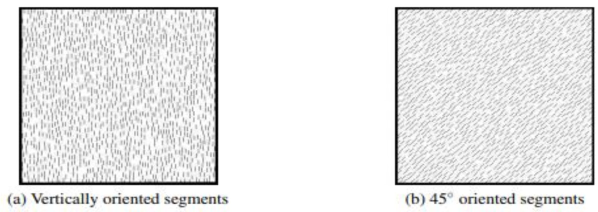
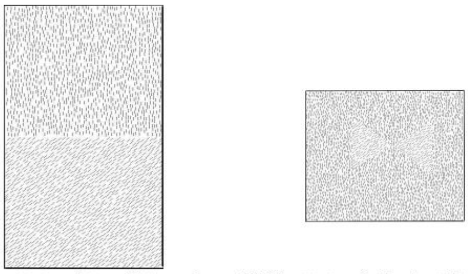


Figure.11. Synthesized images for texture detection with oriented line segments as the texture elements

Figure.11 displays the artificially created 400 x 400 pixel pictures with a single textured section. Line segments with a 45° orientation make up the textures in Figure 11b, whereas vertical line segments do so in Figure 11a. The picture was produced by randomly placing the oriented line segments in pre-existing, all-white backdrop pixels. We produced synthetic pictures with two distinct textured areas as shown in Figure.12 to evaluate the texture recognition and segmentation with two different textures. With our modified version of the algorithm, we were able to distinguish between two distinct textures in these photographs.

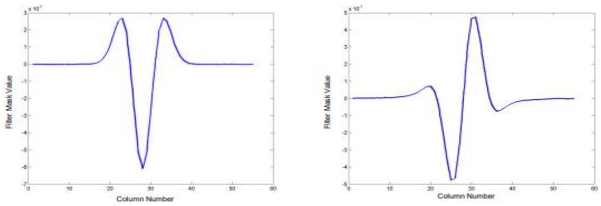


(a) Different textures in top and bottom region (b) Different textures inside and outside the polygon

Figure.12. Synthesized image with two different texture elements

6. Proposed method and Experiments

For texture identification, oriented Gaussian Derivative filters are utilised, as described in [17]. We began by using the texture detection technique from [17]. Following are the stages for the algorithm that identifies k different textures in the image:



(a) Even symmetric filter (b) Odd symmetric filter

Figure.13. For vertically oriented even and odd symmetric filters, the filter mask values are arranged in a single row. The filter response produced on a uniform picture patch (with the same value for all pixels in the patch) will be zero since the mean value of the filter mask is zero.

1.Apply several scales and orientations of Derivative of Oriented Gaussian (DOG) filters to the picture. 2. Concatenate the results of all n filters at each pixel point to form an n-dimensional vector. 3. Group the vectors of filter response into k groups. 4. A texture component, referred to as a texton, in the picture is associated with each cluster centre. 5. Each pixel's cluster label identifies the texture in that area. However, here we suggest to utilise simply the magnitude rather than the filter response coupled with the magnitude and sign. This causes the pixels whose filter response magnitude is closest to zero to cluster. Analyzing the mean and variance of the filter responses allows us to answer the question of the ideal settings. We use a scale that allows for the greatest feasible separation between the mean of the filter response with dominating orientations and other orientations. In the selected range of filter scale values, the variation of the filter response for the dominating orientations shouldn't be excessive.

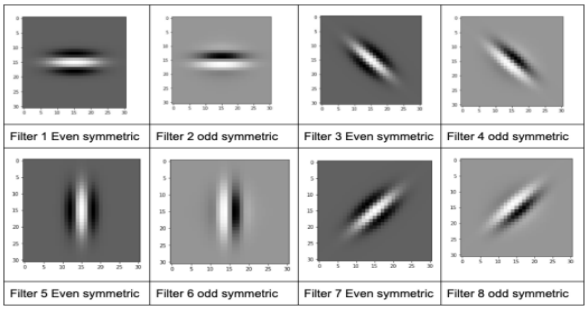
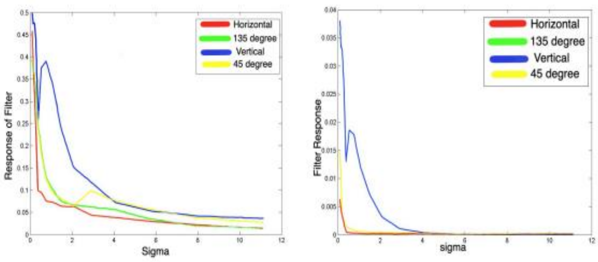


Figure.14. Filter masks with 4 different orientations used to segment the synthetic images

To create filter banks, filter the picture using the filter bank, and cluster the filter response, we utilised the Matlab code from [10]. With each orientation and scale, the algorithm generates filter banks using the oriented Gaussian's second derivative and its Hilbert transform. A pair of Hilbert transform filters creates odd and even symmetric filters that are complimentary to one another for any scale and orientation. Together, these two filters are capable of capturing all intensity gradient fluctuations at a certain scale and direction. Figure.14 displays the filter masks that were used to segment the synthetic pictures. At a single scale, we employed filters with four distinct orientations. Figure.13 for odd and even symmetric filters illustrates the filter mask values along a single row for the vertically oriented filter. When the intensity is uniform, these filters provide no reaction. The filter responds if there is an intensity fluctuation, such as an edge, at either the positive or negative peaks of the filter mask. Whether the edge coincides with the filter mask's positive or negative peak determines the sign of the filter response.

7. Results and Discussions

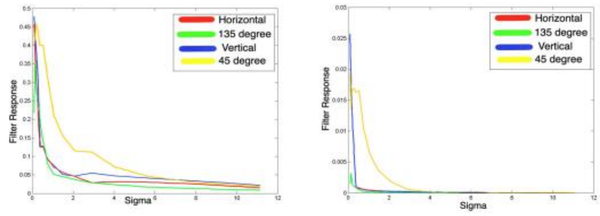
Figures 15a and 15b, respectively, show the mean and variance of filter responses as a function of for input pictures. We anticipate that the filter placed vertically will provide the greatest response for the picture in Figure 12. Therefore, from the standpoint of texture detection, we are searching for a that will provide a bigger response for filters that are vertically oriented as opposed to filters that are oriented in any other manner. In order to guarantee that the filter responds well to most of the spots in the picture, we also demand that the variance of vertically oriented filters be as low as feasible for this image. Our analysis of mean and variance graphs leads us to the conclusion that the ideal value of is close to 2. A good requirement is when the response of a vertically oriented filter is much bigger than that of any other orientation for less than 2. However, for small sigma, the variation of the filter response is greater. As a result, we decide that the ideal scale for recognising these textures is =2.



(a) Mean of filter response (b) Variance of filter response

Figure.15. Filter response average and variance for the picture in Figure (vertically oriented). The average of the scalar values for the filter response at each position in the filtered picture is the mean filter response. The variance is calculated similarly for each position in the filtered picture. Response for even symmetric filters is shown in the graph. The behaviour of odd symmetric filters exhibited a similar pattern.

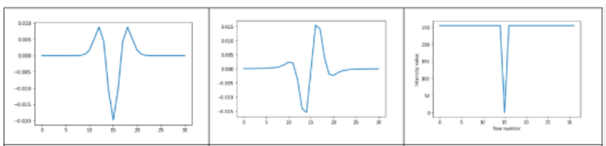
Figure.16a and Figure.16b, respectively, exhibit the mean and variance of filter responses as a function of for input pictures in Figure. We anticipate that the filter positioned vertically will provide the highest response for the picture in Figure 13. Therefore, from the standpoint of texture detection, we are searching for a that will provide a bigger response for filters that are vertically oriented as opposed to filters that are oriented in any other manner. In order to guarantee that the filter responds well to most of the spots in the picture, we also demand that the variance of vertically oriented filters be as low as feasible for this image. Our analysis of mean and variance graphs leads us to the conclusion that the ideal value of is close to 2. A good requirement is when the response of a vertically oriented filter is much bigger than that of any other orientation for less than 2. However, for small sigma, the variation of the filter response is greater. As a result, we decide that the ideal scale for recognising these textures is =2.



(a) Mean of filter response (b) Variance of filter response

Figure.16. Mean and Variance of Filter Response for 45° Oriented Image in Figure FC12b. The average of the scalar values for the filter response at each position in the filtered picture is the mean filter response. The variance is calculated similarly for each position in the filtered picture. Response for even symmetric filters is shown in the graph. The behaviour of odd symmetric filters exhibited a similar pattern.

Figure's textures have white backgrounds with black lines as accents. Figure.17 displays an example of a single row of intensity values for an image patch with a size of 32 by 32, an even symmetric filter mask, and an odd symmetric filter mask. Since the image row has a constant value throughout except for the centre, where it is zero, it is obvious from the picture that the dot product of the odd filter, shown in Figure 17b, and the image row, shown in Figure 17c, is zero. The odd symmetric filter weights are designed in such a way that the weighted average is 0 at any location where the intensity value of the picture row is constant.



a) Single Row of Even symmetric filter mask	b) Single row of Odd symmetric filter mask	c) Intensity values corresponding to a single row of image patch with black vertical line (Intensity value =0) in white back ground (intensity value 255) as texture

Figure.17. An example of a single row of intensity values for a 32x32 picture patch, an even symmetric filter mask, an odd symmetric filter mask, and both

On the other hand, since the image intensity is zero at the negative peak of the filter mask, the even symmetric filter generates a net positive response.

8. Conclusion

The technique of drawing out important information from photos is known as image analysis. Three main categories of image analysis were taken into consideration in this study: picture classification, object identification, and image segmentation. While object detection refers to the process of locating the rectangular borders of distinct things in the picture, image classification refers to the process of assigning k different labels to an image. This study focuses on the recognition and representation of texture, a crucial aspect of picture analysis. Utilizing a well-known texture identification technique built on filter banks, we conduct tests on a variety of artificial and real-world photos. We pinpoint certain particular situations in which the algorithm fails and suggest a change to the original approach that produces better segmentations.

References

1. C. Cortes and V. Vapnik, "Support-vector networks," Machine learning, vol. 20, no. 3, pp. 273–297, 1995. doi:https://doi.org/10.1007/BF00994018.
2. G. I. Webb, Na¨ıve Bayes, pp. 713–714. Boston, MA: Springer US, 2010. doi:ht tps://doi.org/10.1007/978-0-387-30164-8 576.
3. A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," Journal of the Royal Statistical Society: Series B (Methodological), vol. 39, no. 1, pp. 1–22, 1977. doi:https://doi.org/10.1111/j.2517-6161.1977.tb01600.x.
4. S. Lloyd, "Least squares quantization in pcm," IEEE transactions on information theory, vol. 28, no. 2, pp. 129–137, 1982. doi:https://doi.org/10.1109/TIT.1982.105 6489.
5. M. Hossain, "Whitening and coloring transforms for multivariate gaussian random variables," Project Rhea, vol. 3, 2016. Available:https://www.projectr hea.org/rhea/images/1/15/Slecture ECE662 Whitening and Colorin g Transforms S14 MH.pdf.
6. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol. 1, pp. 886–893, IEEE, 2005. doi:https://doi.org/10.1109/CVPR.2005.17 7.
7. D. G. Lowe, "Object recognition from local scale-invariant features," in Proceedings of the seventh IEEE international conference on computer vision, vol. 2, pp. 1150–1157, IEEE, 1999. doi:https://doi.org/10.1109/ICCV.1999.79 0410.
8. F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in 2007 IEEE conference on computer vision and pattern recognition, pp. 1–8, IEEE, 2007. doi:https://doi.org/10.1109/CVPR.2007.38 3266.
9. F. Perronnin, J. Sanchez, and T. Mensink, "Improving the fisher kernel for large-´ scale image classification," in European conference on computer vision, pp. 143–156, Springer, 2010. doi:https://doi.org/10.1007/978-3-642-15561- 1 11.
10. M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3606–3613, 2014. doi:https://doi.org/10.1109/CVPR.2014.46 1.
11. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015. doi:https://doi.org/10.1007/s11263-015-0816-y.
12. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255, IEEE, 2009.

doi:https://doi.org/10.1109/CVPR.2009.5206848.
13. G. A. Miller, "Wordnet: a lexical database for english," Communications of the ACM, vol. 38, no. 11, pp. 39–41, 1995. doi:https://doi.org/10.1145/219717.219748.
14. A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar, "Panop- ´ tic segmentation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9404–9413, 2019. doi:https://doi.org/10.1109/CVPR.2019.00963.
15. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012. Available:https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper. pdf.
16. J. Malik, S. Belongie, J. Shi, and T. Leung, "Textons, contours and regions: Cue integration in image segmentation," in Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 918–925, IEEE, 1999. doi:https://doi.org/10.1109/ICCV.1999.790346.
17. J. Shi and J. Malik, "Normalized cuts and image segmentation," IEEE Transactions on pattern analysis and machine intelligence, vol. 22, no. 8, pp. 888–905, 2000. doi:https://doi.org/10.1109/34.868688.
18. W. Tao, H. Jin, and Y. Zhang, "Color image segmentation based on mean shift and normalized cuts," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 37, no. 5, pp. 1382–1389, 2007. doi:https://doi.org/10.1109/TSMCB.2007.902249.
19. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004. doi:https://doi.org/10.1023/B:VISI.0000029664.99615.94.
20. V. Vapnik, "Principles of risk minimization for learning theory," in Advances in neural information processing systems, pp. 831–838, 1992. Available:https://dl.acm.org/doi/10.5555/2986916.2987018.
21. S. K. Vengalil, N. Sinha, S. S. Kruthiventi, and R. V. Babu, "Customizing cnns for blood vessel segmentation from fundus images," in 2016 International Conference on Signal Processing and Communications (SPCOM), pp. 1–4, IEEE, 2016. doi:https://doi.org/10.1109/SPCOM.2016.7746702.
22. S. K. Vengalil, N. Sinha, and G. S. Raghavan, "Modified oriented gaussian derivative filter based texture detection algorithm and parameter estimation," in 2015 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), pp. 1–6, IEEE, 2015. doi:https://doi.org/10.1109/CONECCT.2015.7383920.
23. R. Lahoti, S. K. Vengalil, P. B. Venkategowda, N. Sinha, and V. V. Reddy, "Whole tumor segmentation from brain mr images using multi-view 2d convolutional neural network," in 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp. 4111–4114, IEEE, 2021. doi:https://doi.org/10.1109/EMBC46164.2021.9631035.
24. D. Bhattacharya, S. K. Vengalil, N. Sinha, J. Saini, and V. Santosh, "Classification of grade-iii astrocytomas based on idh mutation using lbp texture analysis from structural mris," in ICVGIP WORKSHOP ON MEDICAL IMAGE PROCESSING (MedImage 2018), Springer, 2018. Available:https://cloud.iitmandi.ac.in/d/414f49883a/files/?p=%2FMedImage 3.pdf.
25. D. Bhattacharya, S. K. Vengalil, N. Sinha, J. Saini, P. Pal, and M. Sandhya, "Structural mri based texture analysis of corpus callosum in patients with progressive supraneuclear palsy," in TENCON 2019- 2019 IEEE Region 10 Conference (TENCON), pp. 441–446, IEEE, 2019. doi:https://doi.org/10.1109/TENCON.2019.8929403.
26. S. K. Vengalil, K. Xavier, K. A. Sai, S. Harsha, G. Barma, and N. Sinha, "A hybrid approach

for table detection in document images," in 5th Workshop on Document Analysis and Recognition In Conjunction with ICVGIP 2021 (DAR2021), Springer, 2021. Available:https://wdar21.github.io/.

27. S. K. Vengalil and N. Sinha, "Semi-supervised learning using variational autoencoder - a cluster based approach," in Pattern Recognition and Machine Intelligence: 9th International Conference (PReMI 2021), Springer, 2021. Available:https://www.isical.ac.in/~premi21/.

28. H. Jegou, M. Douze, C. Schmid, and P. P ´ erez, "Aggregating local descriptors ´ into a compact image representation," in 2010 IEEE computer society conference on computer vision and pattern recognition, pp. 3304–3311, IEEE, 2010. doi:https://doi.org/10.1109/CVPR.2010.5540039.

29. B. Zhang and Z. et al., "Retinal vessel extraction by matched filter with firstorder derivative of gaussian," Computers in biology and medicine, vol. 40, no. 4, pp. 438–445, 2010. doi:https://doi.org/10.1016/j.compbiomed.2010.02.008.

30. Z. Yavuz and C. Kose, "Retinal blood vessel segmentation using gabor ¨ filter and top-hat transform," in 2011 IEEE 19th Signal Processing and Communications Applications Conference (SIU), pp. 546–549, IEEE, 2011. doi:https://doi.org/10.1109/SIU.2011.5929708.

eISSN 1303-5150