Dr. D. Y. Patil Pratishthan's

DR. D. Y. PATIL INSTITUTE OF ENGINEERING, MANAGEMENT & RESEARCH

Approved by A.I.C.T.E, New Delhi , Maharashtra State Government, Affiliated to Savitribai Phule Pune University
Sector No. 29, PCNTDA , Nigidi Pradhikaran, Akurdi, Pune 411044. Phone: 020–27654470, Fax: 020-27656566
Website :www.dypiemr.ac.in Email : principal.dypiemr@gmail.com

---

Department of Artificial Intelligence and Data Science

LAB MANUAL
Elective I Laboratory Pattern Recognition (TE) Semester I

Prepared by:
Ms. Priyanka Abhale

Elective I Laboratory

Course Code	Course Name	Teaching Scheme (Hrs./ Week)	Credits
317525	Elective I Laboratory Pattern Recognition	2	01

Course Objectives:

- To understand fundamentals of pattern recognition.
- To Study syntactic approach in pattern recognition.
- To study statistical approaches in pattern recognition.
- To study artificial neural network-based pattern recognition

Course Outcomes:

On completion of the course, learner will be able to–

- CO1: Apply statistical pattern recognition approaches.
- CO2: Implement different approaches of syntactic pattern recognition.
- CO3: Develop artificial neural network-based pattern recognition system.

Table of Contents

Group A

Lab Assignment No.	1
Title	Use Bayesian Decision theory of statistical pattern recognition to classify the object
Roll No.	
Class	TE
Date of Completion	
Subject	**Elective I Laboratory Pattern Recognition**
Assessment Marks	
Assessor's Sign	

ASSIGNMENT No: 01

Title: Use Bayesian Decision theory of statistical pattern recognition to classify the object

Problem Statement: Use Bayesian Decision theory of statistical pattern recognition to classify the object

Objective:

- Understand and implement Bayesian Decision theory
- Analyze statistical pattern recognition

Outcomes:

- Ability to apply Bayesian Decision theory for problem-solving method and knowledge representation technique.

Software Required:

- Python

Theory:

Bayesian decision theory refers to the statistical approach based on tradeoff quantification among various classification decisions based on the concept of Probability (Bayes Theorem) and the costs associated with the decision.

It is basically a classification technique that involves the use of the Bayes Theorem which is used to find the conditional probabilities.

In **Statistical Pattern Recognition**, we will focus on the statistical properties of patterns that are generally expressed in probability densities **(pdf's and pmf's)**, and this will command most of our attention in this article and try to develop the fundamentals of the Bayesian decision theory.

A random variable is a function that maps a possible set of outcomes to some values like while tossing a coin and getting head H as 1 and Tail T as 0 where 0 and 1 are random variables.

Bayes Theorem

The conditional probability of A given B, represented by P(A | B) is the chance of occurrence of A given that B has occurred.

P(A | B) = P(A,B)/P(B) or

By Using the Chain rule, this can also be written as:

P(A,B) = P(A|B)P(B)=P(B|A)P(A)

P(A | B) = P(B|A)P(A)/P(B) ——- (1)

Where, **P(B) = P(B,A) + P(B,A') = P(B|A)P(A) + P(B|A')P(A')**

Here, equation (1) is known as the **Bayes Theorem of probability**

Our aim is to explore each of the components included in this theorem. Let's explore step by step:

(a) Prior or State of Nature:

- Prior probabilities represent how likely is each Class is going to occur.
- Priors are known before the training process.
- The state of nature is a random variable **P(w_i)**.
- If there are only two classes, then the sum of the priors is **P(w_1) + P(w_2)=1**, if the classes are exhaustive.

(b) Class Conditional Probabilities:

- It represents the probability of how likely a feature x occurs given that it belongs to the particular class. It is denoted by, **P(X|A)** where x is a particular feature
- It is the probability of how likely the feature x occurs given that it belongs to the class w_i.
- Sometimes, it is also known as the **Likelihood**.
- It is the quantity that we have to evaluate while training the data. During the training process, we have input(features) X labeled to corresponding class w and we figure out the likelihood of occurrence of that set of features given the class label.

(c) Evidence:

- It is the probability of occurrence of a particular feature i.e. **P(X)**.
- It can be calculated using the chain rule as, $\mathbf{P(X) = \Sigma_{in} P(X \mid w_i)\, P(w_i)}$
- As we need the likelihood of class conditional probability is also figure out evidence values during training.

(d) Posterior Probabilities:

- It is the probability of occurrence of Class A when certain Features are given
- It is what we aim at computing in the test phase in which we have testing input or features (the given entity) and have to find how likely trained model can predict features belonging to the particular class w_i.

For a better understanding of the above theory, we consider an example

Problem Description

Suppose we have a classification problem statement where we have to classify among the object-1 and object-2 with the given set of features $\mathbf{X = [x_1, x_2, \ldots, x_n]^T}$.

Objective

The main objective of designing a such classifier is to suggest actions when presented with unseen features, i.e, object not yet seen i.e, not in training data.

In this example let w denotes the state of nature with $\mathbf{w = w_1}$ **for object-1** and $\mathbf{w = w_2}$ **for object-2**. Here, we need to know that in reality, the state of nature is so unpredictable that we generally consider that was variable that is described probabilistically.

Priors

- Generally, we assume that there is some prior value $P(w_1)$ that the next object is object-1 and $P(w_2)$ that the next object is object-2. If we have no other object as in this problem then the sum of their prior is 1 i.e. the priors are exhaustive.
- The prior probabilities reflect the prior knowledge of how likely we will get object-1 and object-2. It is domain-dependent as the prior may change based on the time of year they are being caught.

It sounds somewhat strange and when judging multiple objects (as in a more realistic scenario) makes this decision rule stupid as we always make the same decision based on the largest prior even though we know that any other type of objective also might appear governed by the leftover prior probabilities (as priors are exhaustive in nature).

Consider the following different scenarios:

- If **$P(\omega_1)$>>> $P(\omega_2)$**, our decision in favor of ω_1 will be correct most of the time we predict.
- But if **$P(\omega_1)$= $P(\omega_2)$**, half probable of our prediction of being right. In general, the probability of error is the minimum of $P(\omega_1)$ and $P(\omega_2)$, and later in this article, we will see that under these conditions no other decision rule can yield a larger probability of being correct.

Feature Extraction process (Extract feature from the images)

A suggested set of features- **Length**, **width**, **shapes of an object**, etc.

In our example, we use the **width x**, which is more **discriminatory** to improve the decision rule of our classifier. The different objects will yield different variable-width readings and we usually see this variability in probabilistic terms and also we consider x to be a continuous random variable whose distribution depends on the type of object **w_j**, and is expressed as $p(x|\omega_j)$ (probability distribution function pdf as a continuous variable) and known as the class-conditional probability density function. Therefore,

The pdf $p(x|\omega_1)$ is the probability density function for feature x given that the state of nature is ω_1 and the same interpretation for $p(x|w_2)$.

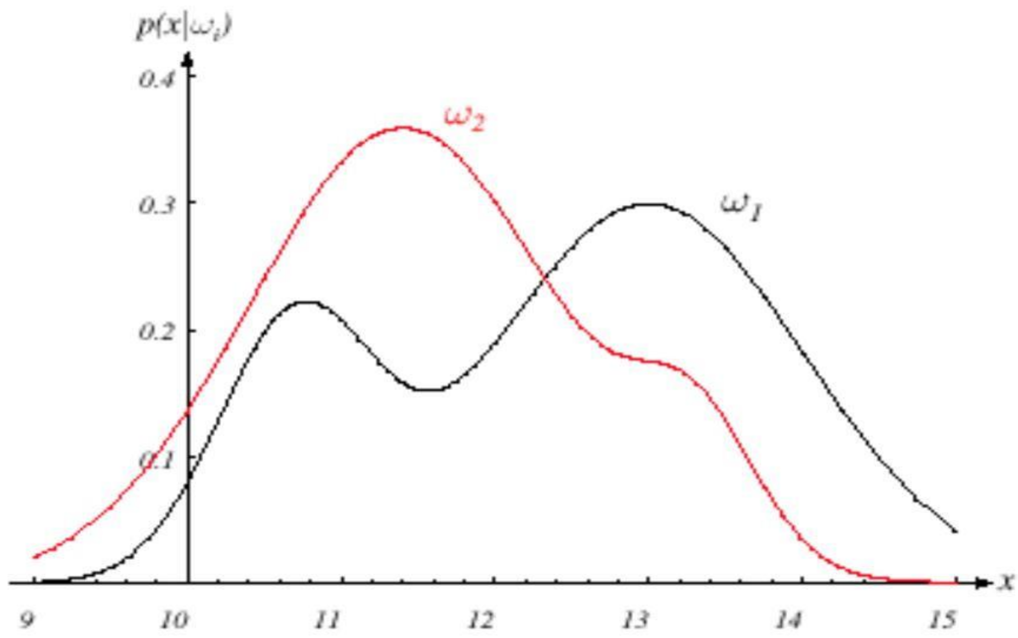


Fig. Picture Showing pdf for both classes

Suppose that we are well aware of both the prior probabilities $P(\omega_j)$ and the conditional densities $p(x|\omega_j)$. Now, we can arrive at the Bayes formula for finding posterior probabilities:

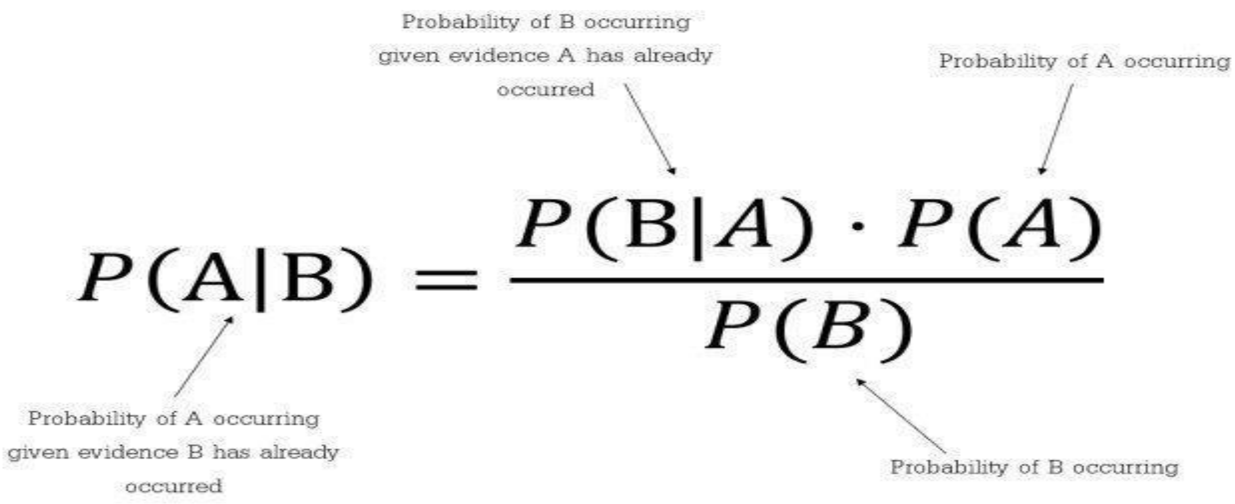


Fig. Formula of Bayes Theorem

Bayes' formula gives us intuition that by observing the measurement of x we can convert the prior $P(\omega_j)$ to the posteriors, denoted by $P(\omega_j|x)$ which is the probability of ω_j given that feature value x has been measured.

$p(x|\omega_j)$ is known as the likelihood of ω_j with respect to x.

The evidence factor, p(x), works as merely a scale factor that guarantees that the posterior probabilities sum up to one for all the classes.

Bayes' Decision Rule

The decision rule given the posterior probabilities is as follows

If $\mathbf{P(w_1|x)} > \mathbf{P(w_2|x)}$ we would decide that the object belongs to class w_1, or else class w_2.

Probability of Error

To justify our decision we look at the probability of error, whenever we observe x, we have,

P(error|x)= $P(w_1|x)$ if we decide w_2, and $P(w_2|x)$ if we decide w_1

As they are exhaustive and if we choose the correct nature of an object by probability P then the leftover probability (1-P) will show how probable is the decision that it the not the decided object.

We can minimize the probability of error by deciding the one which has a greater posterior and the rest as the probability of error will be minimum as possible. So we finally get,

$P(error|x) = \min [P(\omega_1|x),P(\omega_2|x)]$

And our Bayes decision rule as,

Decide ω_1 if $P(\omega_1|x) > P(\omega_2|x)$; otherwise decide ω_2

This type of decision rule highlights the role of the posterior probabilities. With the help Bayes theorem, we can express the rule in terms of conditional and prior probabilities.

The evidence is unimportant as far as the decision is concerned. As we discussed earlier it is working as just a scale factor that states how frequently we will measure the feature with value x; it assures **$P(\omega_1|x)+ P(\omega_2|x) = 1$**.

So by eliminating the unrequired scale factor in our decision rule we have, the similar decision rule by Bayes theorem as,

Decide ω_1 if $p(x|\omega_1)P(\omega_1) > p(x|\omega_2)P(\omega_2)$; otherwise decide ω_2

Now, let's consider 2 cases:

- **Case-1:** If class conditionals are equal i.e, **$p(x|\omega_1)= p(x|\omega_2)$**, then we arrive at our premature decision rule governed by just priors.
- **Case-2:** On the other hand, if priors are equal i.e, **$P(\omega_1)= P(\omega_2)$** then the decision is entirely based on class conditionals $p(x|\omega_j)$.

This completes our example formulation!

Generalization of the preceding ideas for Multiple Features and Classes

Bayes classification: Posterior, likelihood, prior, and evidence

$P(w_i \mid X) = P(X \mid w_i)\, P(w_i) / P(X)$

Posterior = Likelihood* Prior/Evidence

We now discuss those cases which have multiple features as well as multiple classes,

Let the Multiple Features be $\mathbf{X_1, X_2, \ldots X_n}$ and Multiple Classes be $\mathbf{w_1, w_2, \ldots w_n}$, then:

$P(w_i \mid X_1, \ldots. X_n) = P(X_1,\ldots., X_n|w_i)*P(w_i)/P(X_1,\ldots X_n)$

Where,

Posterior = $P(w_i \mid X_1, \ldots. X_n)$

Likelihood = $P(X_1,\ldots., X_n|w_i)$

Prior = $P(w_i)$

Evidence = $P(X_1,\ldots ,X_n)$

In cases of the same incoming patterns, we might need to use a drastically different cost function, which will lead to different actions altogether. Generally, different decision tasks may require features and yield boundaries quite different from those useful for our original categorization problem.

Conclusion: Thus, we have studied Bayesian Decision theory of statistical pattern recognition to classify the object

Lab Assignment No.	2
Title	Implement Cocke–Younger–Kasami (CYK) Parsing Algorithm using Syntactic Pattern Recognition
Roll No.	
Class	TE
Date of Completion	
Subject	**Elective I Laboratory Pattern Recognition**
Assessment Marks	
Assessor's Sign	

ASSIGNMENT No: 02

Title: Implement Cocke–Younger–Kasami (CYK) Parsing Algorithm using Syntactic Pattern Recognition

Problem Statement: Implement Cocke–Younger–Kasami (CYK) Parsing Algorithm using Syntactic Pattern Recognition

Objective:

- Understand and implement Cocke–Younger–Kasami (CYK) Parsing Algorithm
- Analyze Syntactic pattern recognition

Outcomes:

- Ability to apply Bayesian Decision theory for problem-solving method and knowledge representation technique.

Software Required:

- Python

Theory:

Grammar denotes the syntactical rules for conversation in natural language. But in the theory of formal language, grammar is defined as a set of rules that can generate strings. The set of all strings that can be generated from a grammar is called the language of the grammar.

Context Free Grammar:

We are given a Context Free Grammar G = (V, X, R, S) and a string w, where:

- *V* is a finite set of variables or non-terminal symbols,
- *X* is a finite set of terminal symbols,
- *R* is a finite set of rules,
- *S* is the start symbol, a distinct element of *V*, and
- *V* and *X* are assumed to be disjoint sets.
-

The **Membership problem** is defined as: Grammar G generates a language L(G). Is the given string a member of L(G)?

Chomsky Normal Form:

A Context Free Grammar G is in Chomsky Normal Form (CNF) if each rule if each rule of *G* is of the form:

- *A* –> *BC*, [with at most two non-terminal symbols on the RHS]
- *A* –> *a*, or [one terminal symbol on the RHS]
- *S* –> nullstring, [null string]

Cocke-Younger-Kasami Algorithm

It is used to solves the membership problem using a dynamic programming approach. The algorithm is based on the principle that the solution to problem *[i, j]* can constructed from solution to subproblem *[i, k]* and solution to sub problem *[k, j]*.

The algorithm requires the Grammar *G* to be in Chomsky Normal Form (CNF).

Note that any Context-Free Grammar can be systematically converted to CNF.

This restriction is employed so that each problem can only be divided into two subproblems and not more – to bound the time complexity.

How does the CYK Algorithm work?

For a string of length **N**, construct a table **T** of size **N x N**. Each cell in the table **T[i, j]** is the set of all constituents that can produce the substring spanning from position i to j. The process involves filling the table with the solutions to the subproblems encountered in the bottom-up parsing process. Therefore, cells will be filled from left to right and bottom to top.

	1	**2**	**3**	**4**	**5**
1	[1, 1]	[1, 2]	[1, 3]	[1, 4]	[1, 5]
2		[2, 2]	[2, 3]	[2, 4]	[2, 5]
3			[3, 3]	[3, 4]	[3, 5]
4				[4, 4]	[4, 5]
5					[5, 5]

Conclusion: Thus, we have studied Cocke–Younger–Kasami (CYK) Parsing Algorithm using Syntactic Pattern Recognition

Group B

Lab Assignment No.	03
Title	Generate a Pattern from String using syntactical Pattern Approach
Roll No.	
Class	TE
Date of Completion	
Subject	Software Laboratory I
Assessment Marks	
Assessor's Sign	

ASSIGNMENT No: 03

Title: Generate a Pattern from String using syntactical Pattern Approach

Problem Statement: To generate a Pattern from String using syntactical Pattern Approach

Objective:

- Understand and implement pattern from string.
- Analyze syntactical pattern approach.

Outcomes:

- Ability to choose an appropriate problem-solving method and knowledge representation technique.

Software Required:

- Python

Theory:

The principal objective of this experiment is to provide an introduction to basic concepts of syntactic pattern recognition. The theory of automata and formal languages is one of the principal elements in the study of digital machines and their processing capabilities. Syntactic pattern recognition employs this theory in innovative ways to develop pattern recognition approaches that are based on knowledge of the underlying structure of pattern classes. This chapter presents also the fundamentals of formal language and automata theory as they apply to pattern recognition.

3.1 Basics Pattern recognition techniques are among the most important tools used in the field of machine intelligence. Pattern recognition can be defined as the categorization of input data into identifiable classes via the extraction of significant features or attributes of the data from a background of irrelevant detail C1353. A pattern is essentially an arrangement. It may be defined as a quantitative or structural description of an object or some other entity of interest. A pattern class is a set of patterns that share some common properties. The subject matter of 29 pattern recognition by machine deals with techniques for assigning patterns to their respective classes, automatically and with as little human intervention as possible.

The study of pattern recognition problems may be logically divided into two major categories:

1. The study of the pattern recognition capability of human beings and other living organisms.

2. The development of underlying theory and practical techniques for machine implementation of a given recognition task. The first area falls in the domain of psychology, physiology and biology while the second area is in the domain of engineering, computer science and applied mathematics.

Approaches to pattern recognition system design may be divided into two principal categories:

- the decision theoretic approach ; and

- the syntactic approach.

The decision-theoretic approach is based on the utilization of decision functions for classifying pattern vectors and is ideally suited for applications where patterns can have meaningful representation in vector form. There are applications, however, where the structure of a pattern plays an important role in the classification process. In these situations, the decision theoretic approach has serious drawbacks because it lacks a suitable formalism for handling pattern structures and their relationships. For example, the decision—theoretic approach finds few applications to ECG analysis, since in this case the structure and relationships of the various components of ECG are of fundamental importance in establishing a meaningful recognition scheme.

The syntactic approach to pattern recognition has been receiving increased attention during the past few years because it possesses the struiT>; ?-hand1ing capability lacked by the decision-theoretic approach. Syntactic pattern recognition is based on concepts from formal language theory, the origins of which may be traced to the middle 1950s with the development of mathematical models of grammars by Noam Chomsky.

Basic to the syntactic pattern recognition approach is the deccuifiosition of patterns into sub patterns or primitives. By tracking a complex pattern it is possible to detect and encode the primitives in the form of a string of qualifiers. Suppose that we interpret each primitive as being a symbol permissible in some grammar, where a grammar is a set of rules of syntax for the generation of sentences from the given symbols. Once the grammar has been established, the syntactic pattern recognition process is, in principle, straight forward. Given a sentence representing an input pattern, the problem is to decide whether the input pattern represents a valid sentence. If a pattern is not a sentence of the language under consideration, it is assigned to a rejection class. 3.2 String grammars and language

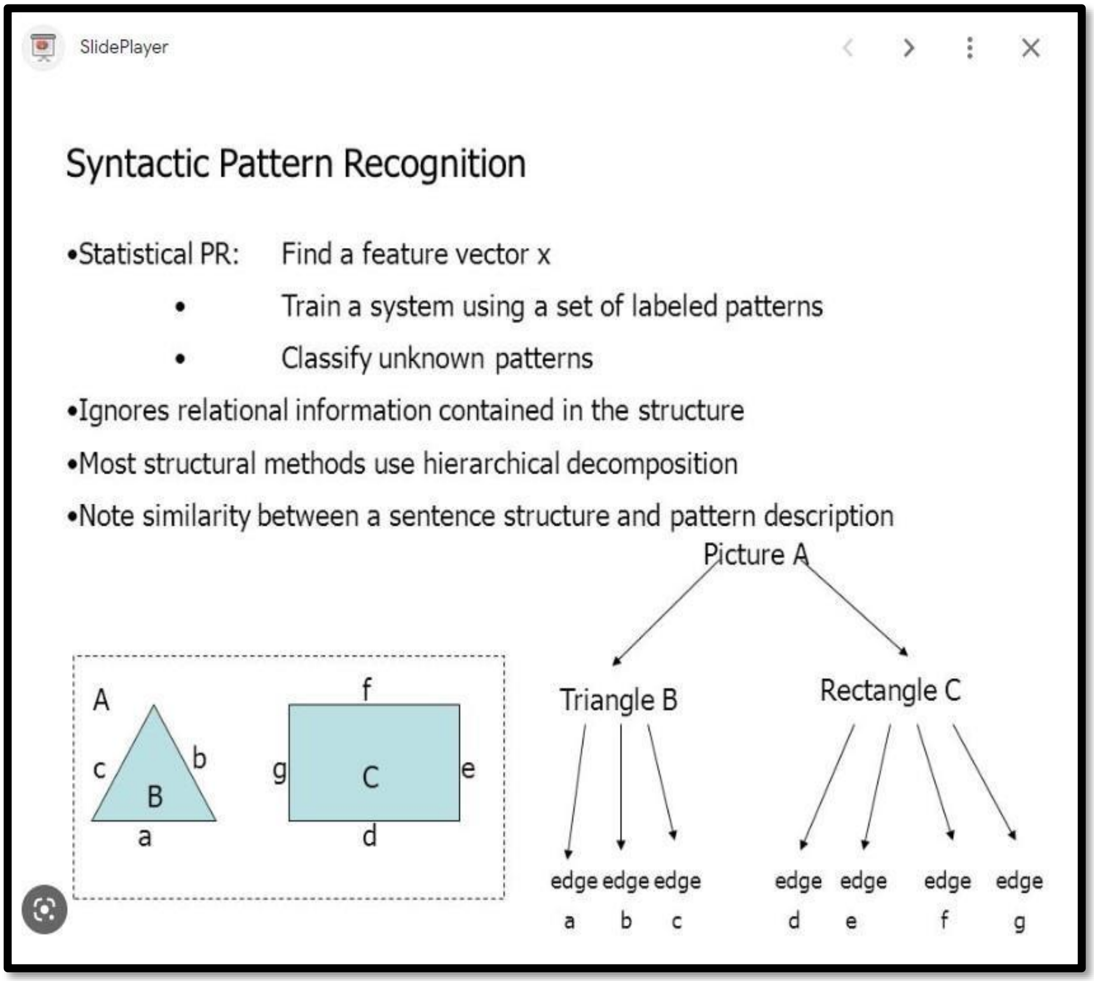
As indicated in section 3.1, the methods of syntactic pattern recognition are based on mathematical systems in which the patterns of a class are represented as elements of a language. One of the principal requirements in designing a syntactic pattern recognition system is the development of a grammar capable of generating a given class of patterns. Most of the material presented here deals with the definition and interpretation of 31 grammars suitable for pattern representation and recognition.

The following concise definitions establish some of the principal notation and concepts used in formal language theory.

Two categories of sets of interest in formal language theory are finite sets, which have a finite number of elements, and countably infinite sets whose elements can be placed in a one-to-one correspondence with the positive integer.

Given sets A and S, their Cartesian product >4 X S is the collection of all ordered pairs in ^ and & in B . For sets A^ this may be extended to the y^i-fold Cartesian product .4^ X X ...X A^^, which is the set of all ri-tuples , ...) for a.^ in in A . The null set is the set containing no elements. V l ^ A relation from set A to set S is a subset of ^4 X S, i.e., R ^ A X B . If for each element a of A there is exactly one element h of B such that (Start by putting any one of the

-
- In syntactic pattern recognition, **a parser or error correcting parser checks an unknown input whether it is in accordance with the rules of a grammar that describes all members of a pattern class**. This chapter reviews basic concepts and algorithms applied in structural and syntactic pattern recognition.



Conclusion: Thus, we have studied a Pattern from String using syntactical Pattern Approach

Lab Assignment No.	4
Title	Apply suitable pattern recognition technique to perform Character Recognition
Roll No.	
Class	TE
Date of Completion	
Subject	Software Laboratory I
Assessment Marks	
Assessor's Sign	

ASSIGNMENT No: 04

Title: Apply suitable pattern recognition technique to perform Character Recognition

Problem Statement: Implement suitable pattern recognition technique to perform Character Recognition

Objective:

- Understand pattern recognition
- Implement character recognition

Outcome:

- OCR Technology
- optical character recognition work

Software Required: python

Theory: Pattern recognition is the ability of machines to identify patterns in data, and then use those patterns to make decisions or predictions. This guide provides an overview of the most important techniques used to recognize patterns and real-world applications. This article will cover what pattern recognition is, forms of pattern recognition, and analyze forms of pattern recognition in artificial intelligence**.**

Optical Character Recognition (OCR) defines the process of mechanically or electronically converting scanned images of handwritten, typed, or printed text into machine-encoded text. Think of it as the process of turning analog data, digital.

OCR
As OCR stands for optical character recognition, OCR technology deals with the problem of recognizing all kinds of different characters. Both handwritten and printed characters can be recognized and converted into a machine-readable, digital data format.

Think of any kind of serial number or code consisting of numbers and letters that you need digitized. By using OCR you can transform these codes into a digital output. The technology makes use of many different techniques. Put simply, the image taken is processed, the characters extracted, and are then recognized.

What OCR does not do is consider the actual nature of the object that you want to scan. It simply "takes a look" at the characters that you aim to transform into a digital format. For example, if you scan a word it will learn and recognize the letters, but not the meaning of the word.

How does Optical Character Recognition Work?

Let's have a look at three basic steps of optical character recognition: image pre-processing; character recognition; and the post-processing of the output.

Step 1: Image Pre-Processing in OCR

OCR software often pre-processes images to improve the chances of successful recognition. The aim of image pre-processing is an improvement of the actual image data. In this way, unwanted distortions are suppressed and specific image features are enhanced. These two processes are important for the following steps.

Step 2: Character Recognition in OCR

Character Recognition of License Plates

For the actual character recognition, it is important to understand what "feature extraction" is. When the input data is too large to be processed, only a reduced set of features is selected. The features selected are expected to be the important ones while those that are suspected to be redundant are ignored. By using the reduced set of data instead of the initial large one, the performance is increased.

For the process of OCR, this is important as the algorithm has to detect specific portions or shapes of a digitized image or video stream.

Step 3: Post-Processing in OCR

Post-processing is another error correction technique that ensures the high accuracy of OCR. The accuracy can be further improved if the output is restricted by a lexicon. That way, the algorithm can fall back to a list of words that are allowed to occur in the scanned document for example.

OCR is not only used to identify proper words but can also read numbers and codes. This is useful for identifying long strings of numbers and letters, such as serial numbers used in many industries.

To better deal with different types of input OCR, some providers started to develop specific OCR systems. These systems are able to deal with the special images, and to improve the recognition accuracy, even more, they combined various optimization techniques.

For example, they used business rules, standard expressions, or rich information contained in the color image. This strategy of merging various optimization techniques is called "application-oriented OCR" or "customized OCR". It is used in applications such as business card OCR, invoice OCR, and ID card OCR.

Use Cases for OCR Technology

The possibilities for using optical character recognition software is widespread as OCR can be combined with a broad range of technologies. Here are a few examples of possible use cases including OCR software:

1. Identification Processes in OCR

Machine Readable Zone (MRZ) in a Passport
Passports and IDs have a machine-readable zone (MRZ) that can be scanned. OCR can speed up the process of identifying and registering people. This is useful for security forces at borders or other checkpoints. It can also be used for commercial purposes to increase customer engagement, such as the check-in process within hotels, or the registration process with banks and other businesses.

2. Marketing Campaigns with OCR

Leading brands are making use of OCR to run innovative and engaging campaigns to drive engagement with their customers. Think of all the voucher codes that customers can redeem by typing them in. Or numbers printed on the inside of a bottle cap that you need to collect.

All of these campaigns can make use of OCR by integrating the software, which easily integrates into company websites and apps. That way, they minimize the hurdle of online registration and remove the need for customers to typing in a series of numbers and letters.

Take a look at how PepsiCo uses OCR in one of their marketing campaigns in Turkey to scan voucher codes inside packets of their popular chips like Lays, Ruffles, and Doritos:

IBAN Scanning with OCR

The International Bank Account Number (IBAN) serves to identify bank accounts across borders. The IBAN may come in different lengths and can consist of numbers as well as letters. To ease cross-border transactions banking apps can easily integrate OCR software. That way their customers can scan their IBAN instead of tediously typing it in.

Conclusion: Thus, we have studied suitable pattern recognition technique to perform Character Recognition.

Group C

Lab Assignment No.	05
Title	Develop a system for Handwritten Digit Recognition using Neural Network.
Roll No.	
Class	TE
Date of Completion	
Subject	Software Laboratory I
Assessment Marks	
Assessor's Sign	

ASSIGNMENT No: 05

Title: Develop a system for Handwritten Digit Recognition using Neural Network

Problem Statement: To implement a system for Handwritten Digit Recognition using Neural Network

Objective:

- Study neural network
- Understand and implement Handwritten digit recognition.

Outcomes:

- Ability to choose an appropriate system for Handwritten Digit Recognition using Neural Network

Software Required:

- Python

Theory:

Handwritten digit recognition using MNIST dataset is a major project made with the help of Neural Network. It basically detects the scanned images of handwritten digits.

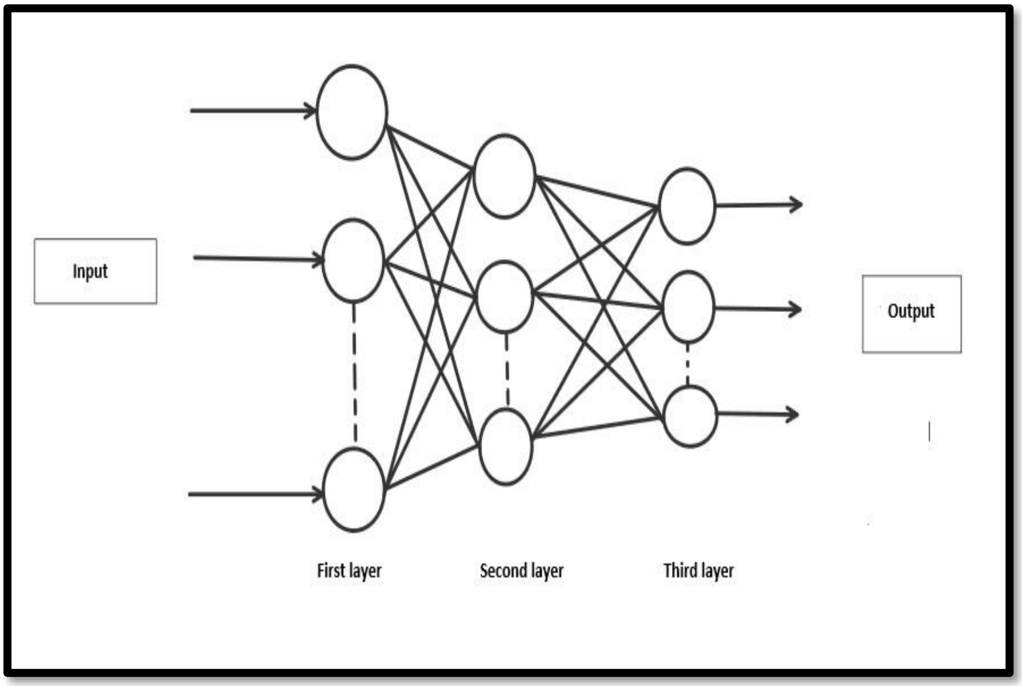
We have taken this a step further where our handwritten digit recognition system not only detects scanned images of handwritten digits but also allows writing digits on the screen with the help of an integrated GUI for recognition.

Approach:
We will approach this project by using a three-layered Neural Network.

The input layer: It distributes the features of our examples to the next layer for calculation of activations of the next layer.
The hidden layer: They are made of hidden units called activations providing nonlinear ties for the network. A number of hidden layers can vary according to our requirements.
The output layer: The nodes here are called output units. It provides us with the final prediction of the Neural Network on the basis of which final predictions can be made.
A neural network is a model inspired by how the brain works. It consists of multiple layers having many activations, this activation resembles neurons of our brain. A neural network tries to learn a set of parameters in a set of data which could help to recognize the underlying relationships. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria.

Methodology:

We have implemented a Neural Network with 1 hidden layer having 100 activation units (excluding bias units). The data is loaded from a .mat file, features(X) and labels(y) were extracted. Then features are divided by 255 to rescale them into a range of [0,1] to avoid overflow during computation. Data is split up into 60,000 training and 10,000 testing examples. Feedforward is performed with the training set for calculating the hypothesis and then backpropagation is done in order to reduce the error between the layers. The regularization parameter lambda is set to 0.1 to address the problem of overfitting. Optimizer is run for 70 iterations to find the best fit model.



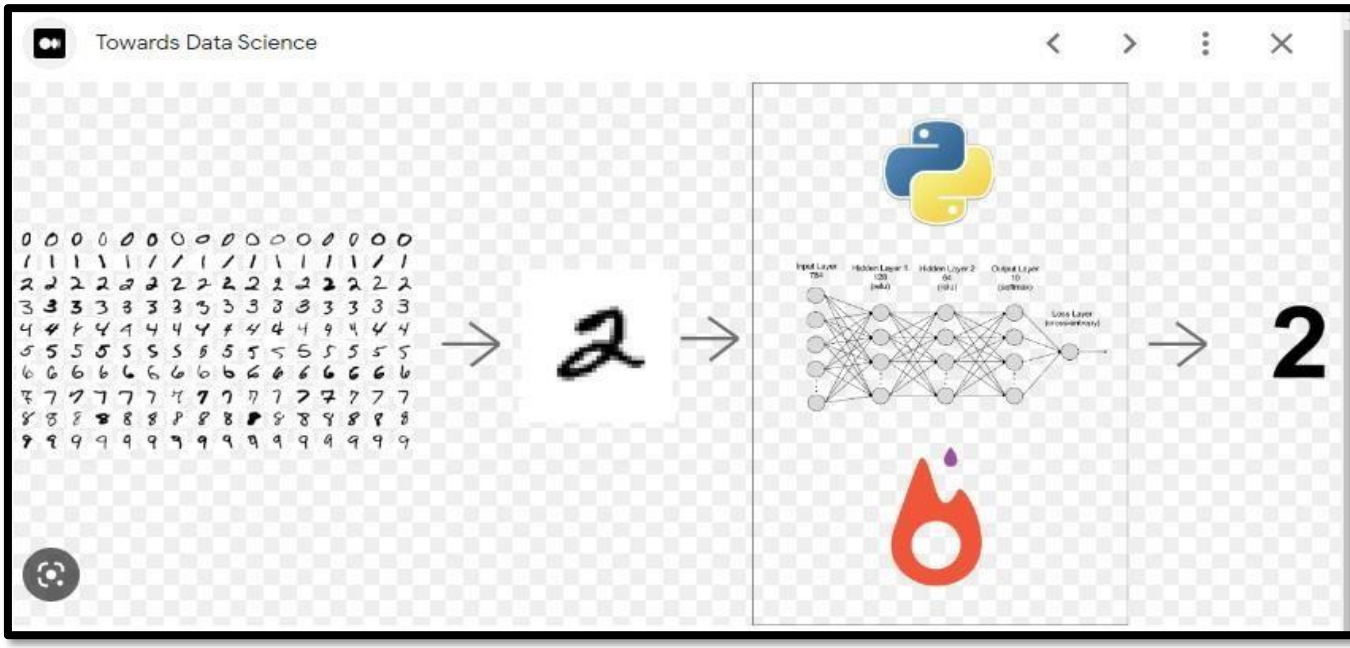


Figure: Neural Network for Recognition of Handwritten Digits

Conclusion: Thus, we have studied Handwritten Digit Recognition using Neural Network.