Q.2 a. The future directions of research & development in AI can be in "Information creation" and "Autonomy". Explain in detail. (8)

Answer: Page 18 of Text Book

b. What do you mean by intelligence? What essential abilities one should possess for intelligence? (8)

Answer: Page 3 of Text Book

Q.3 a. Attempt to unify the following pairs of expression. Either show their most general unifiers or explain why they will not unify. (9)

(i) p(X,Y) and p(a, Z)
(ii) p(X,X) and p(a, b)
(iii) ancestor (X, Y) and ancestor (bill, father (bill)
(iv) ancestor (X, father (X)) and ancestor (david, George)
(v) q(X) and ¬ q(a)
(vi) p(X, a, Y) and p(Z, Z, b)

Answer:

(a) p(X,Y). and p(a,Z). Unifies with { a/X, Y/Z}.

(b) p(X,X). and p(a,b). Fails to unify since both a and b cannot be substituted for X.

(c) ancestor(X,Y). and ancestor(bill, father(bill)). Unifies {bill/X, father(bill)/Y}.

(d) ancestor(X,father(X)). and ancestor(david, george). Unifies only if george is the father(david). Function evaluation should be added to the unification algorithm.

(e) q(X). and ¬ q(a). Will unify if provision is made for ¬ in the unification algorithm.

(f) p(X,a,Y). and p(Z,Z,b). Unifies with {a/X, a/Z, b/Y}.

b. Show that $S \vee R$ is tautologically implied by $(P \vee Q) \wedge (P \rightarrow R) \wedge (Q \rightarrow S)$ (7)

Answer:

P Q R S	(P∨Q)	(P→R)	(Q→S)	(S∨R)	[(P∨Q)∧(P→R)∧(Q→S)] → (S∨R)
0 0 0 0	0	1	1	0	1
0 0 0 1	0	1	1	1	1
0 0 1 0	0	1	1	1	1
0 0 1 1	0	1	1	1	1
0 1 0 0	1	1	0	0	1
0 1 0 1	1	1	1	1	1
0 1 1 0	1	1	0	1	1
0 1 1 1	1	1	1	1	1
1 0 0 0	1	0	1	0	1
1 0 0 1	1	0	1	1	1
1 0 1 0	1	1	1	1	1
1 0 1 1	1	1	1	1	1
1 1 0 0	1	0	0	0	1
1 1 0 1	1	0	1	1	1
1 1 1 0	1	1	0	1	1
1 1 1 1	1	1	1	1	1

Q.4 a. Briefly describe the different techniques for knowledge representation. (8)

Answer:

All of these, in different ways, involve hierarchical representation of data.

 Lists - linked lists are used to represent hierarchical knowledge

 Trees - graphs which represent hierarchical knowledge. LISP, the main programming language of AI, was developed to process lists and trees.

 Semantic networks - nodes and links - stored as propositions.

 Schemas - used to represent commonsense or stereotyped knowledge.

o Frames (Minsky) - Describe objects. Consist of a cluster of nodes and links manipulated as a whole. Knowledge is organised in slots. Frames are hierarchically organised.

o Scripts (Schank and Abelson) - Describe event rather than objects. Consist of stereotypically ordered causal or temporal chain of events.

 Rule-based representations (Newell and Simon) - used in specific problem-solving contexts. Involve production rules containing *if-then* or *situation-action* pairs.

Specific example: problem space representations. Contain:

o Initial state

o Goal state

o Legal operators, i.e. things you are allowed to do

o Operator restrictions, i.e. factors which constrain the application of operators

(More on Problem-space representations and strategies in Semester 2 - Problem solving - expert-novice studies)

 Logic-based representations - may use deductive or inductive reasoning. Contain:

o Facts and premises

o Rules of propositional logic (Boolean - dealing with complete statements

o Rules of predicate calculus (allows use of additional information about objects in the proposition, use of variables and functions of variables

o Measures of certainty - may involve Certainty Factors (eg. If symptom then (CF) diagnosis) which could be derived from expert estimation or from statistical data; Bayesian probability; or Fuzzy logic (in which the concepts or information itself has some associated certainty value).

b. Write a script for "robbing a bank". (8)

Answer: Page 68 of Text Book

Q.5 a. Explain Dempster and Shafer's theory of evidences. (8)

Answer:

The **Dempster–Shafer theory (DST)** is a mathematical theory of evidence. It allows one to combine evidence from different sources and arrive at a degree of belief (represented by a belief function) that takes into account all the available evidence. The theory was first developed by Arthur P. Dempster and Glenn Shafer.

In a narrow sense, the term **Dempster–Shafer theory** refers to the original conception of the theory by Dempster and Shafer. However, it is more common to use the term in the wider sense of the same general approach, as adapted to specific kinds of situations. In particular, many authors have proposed different rules for combining evidence, often with a view to handling conflicts in evidence better.

Dempster–Shafer theory is a generalization of the Bayesian theory of subjective probability; whereas the latter requires probabilities for each question of interest, belief functions base degrees of belief (or confidence, or trust) for one question on the probabilities for a related question. These degrees of belief may or may not have the mathematical properties of probabilities; how much they differ depends on how closely the two questions are related.

Put another way, it is a way of representing epistemic plausibilities but it can yield answers that contradict those arrived at using probability theory.

Often used as a method of sensor fusion, Dempster–Shafer theory is based on two ideas: obtaining degrees of belief for one question from subjective probabilities for a related question, and Dempster's rule for combining such degrees of belief when they are based on independent items of evidence. In essence, the degree of belief in a proposition depends primarily upon the number of answers (to the related questions) containing the proposition, and the subjective probability of each answer. Also contributing are the rules of combination that reflect general assumptions about the data.

In this formalism a **degree of belief** (also referred to as a **mass**) is represented as a **belief function** rather than a Bayesian probability distribution. Probability values are assigned to *sets* of possibilities rather than single events: their appeal rests on the fact they naturally encode evidence in favor of propositions.

Dempster–Shafer theory assigns its masses to all of the non-empty subsets of the entities that compose a system.

Belief and plausibility

Shafer's framework allows for belief about propositions to be represented as intervals, bounded by two values, *belief* (or *support*) and *plausibility*:

belief ≤ *plausibility*.

Belief in a hypothesis is constituted by the sum of the masses of all sets enclosed by it (*i.e.* the sum of the masses of all subsets of the hypothesis). It is the amount of belief that directly supports a given hypothesis at least in part, forming a lower bound. Belief (usually denoted *Bel*) measures the strength of the evidence in favor of a set of propositions.

It ranges from 0 (indicating no evidence) to 1 (denoting certainty). *Plausibility* is 1 minus the sum of the masses of all sets whose intersection with the hypothesis is empty. It is an upper bound on the possibility that the hypothesis could be true, *i.e.* it “could possibly be the true state of the system” up to that value, because there is only so much evidence that contradicts that hypothesis. Plausibility (denoted by *Pl*) is defined to be $Pl(s)=1-Bel(\sim s)$. It also ranges from 0 to 1 and measures the extent to which evidence in favor of $\sim s$ leaves room for belief in *s*. For example, suppose we have a belief of 0.5 and a plausibility of 0.8 for a proposition, say “the cat in the box is dead.” This means that we have evidence that allows us to state strongly that the proposition is true with a confidence of 0.5. However, the evidence contrary to that hypothesis (i.e. “the cat is alive”) only has a confidence of 0.2. The remaining mass of 0.3 (the gap between the 0.5 supporting evidence on the one hand, and the 0.2 contrary evidence on the other) is “indeterminate,” meaning that the cat could either be dead or alive. This interval represents the level of uncertainty based on the evidence in your system.

Hypothesis	Mass	Belief	Plausibility
Null (neither alive nor dead)	0	0	0
Alive	0.2	0.2	0.5
Dead	0.5	0.5	0.8
Either (alive or dead)	0.3	1.0	1.0

The null hypothesis is set to zero by definition (it corresponds to "no solution"). The orthogonal hypotheses "Alive" and "Dead" have probabilities of 0.2 and 0.5, respectively.
This could correspond to "Live/Dead Cat Detector" signals, which have respective reliabilities of 0.2 and 0.5. Finally, the all-encompassing "Either" hypothesis (which simply acknowledges there is a cat in the box) picks up the slack so that the sum of the masses is 1.
The belief for the "Alive" and "Dead" hypotheses matches their corresponding masses because they have no subsets; belief for "Either" consists of the sum of all three masses (Either, Alive, and Dead) because "Alive" and "Dead" are each subsets of "Either". The "Alive" plausibility is 1 − m (Dead) and the "Dead" plausibility is 1 − m (Alive). Finally, the
"Either" plausibility sums m(Alive) + m(Dead) + m(Either). The universal hypothesis ("Either") will always have 100% belief and plausibility—it acts as a checksum of sorts.

b. Write notes on following:
(i) Default Logic **(ii) Fuzzy Logic** **(4×2)**

Answer: Page 111 of Text Book

Q.6 a. Would you use breadth-first or depth-first search for each of the following problems? What would you base your choice on? (8)
(i) A chess playing program.
(ii) A medical diagnostic program.
(iii) A program to determine the best sequence of manufacturing steps to go from raw materials to a finished product.
(iv) A program that attempts to determine if two expressions in the propositional calculus are equivalent.

Answer:
(a) A chess playing program. Has to be depth-first, or at least guided (with heuristics, Chapter 4) depth-first. The branching factor is too large in chess to get to any interesting depth with exhaustive **breadth-first search**.
(b) A medical diagnostic program. Usually done **depth-first**. This allows the program to follow a line of reasoning and only change to a new goal or hypothesis when a previous one is confirmed or denied.

(c) A program to determine the best sequence of manufacturing steps to go from raw materials to a finished product. It should at least have a flavor of **breadth firs**t, if not be totally breadth-first so that some process aren't satisfied to the detriment of later ones.
(d) A program that attempts to determine if two expressions in the propositional calculus are equivalent. This allows unification substitutions to be pushed forward through the remaining expression.

b. Describe Hill climbing search technique and also explain when it will fail. (8)

Answer:
Hill climbing is an optimization technique which belongs to the family of local search. It is relatively simple to implement, making it a popular first choice. Although more advanced algorithms may give better results, in some situations hill climbing works well. Hill climbing can be used to solve problems that have many solutions, some of which are better than others. It starts with a random (potentially poor) solution, and iteratively makes small changes to the solution, each time improving it a little. When the algorithm cannot see any improvement anymore, it terminates. Ideally, at that point the current solution is close to optimal, but it is not guaranteed that hill climbing will ever come close to the optimal solution. For example, hill climbing can be applied to the traveling salesman problem. It is easy tofind a solution that visits all the cities but is be very poor compared to the optimal solution. The algorithm starts with such a solution and makes small improvements to it, such as switching the order in which two cities are visited. Eventually, a much better route is obtained.

Problems with hill climbing: **local maxima** (we've climbed to the top of the hill, And missed the mountain), **plateau** (everything around is about as good as where we are),**ridges** (we're on a ridge leading up, but we can't directly apply an operator to improve our situation, so we have to apply more than one operator to get there)
Solutions include: backtracking, making big jumps (to handle plateaus or poor local maxima), applying multiple rules before testing (helps with ridges).

Q.7 a. Differentiate between:
(i) Data processing and knowledge processing
(ii) Database and knowledge base (8)

Answer: Page 185 of Text Book

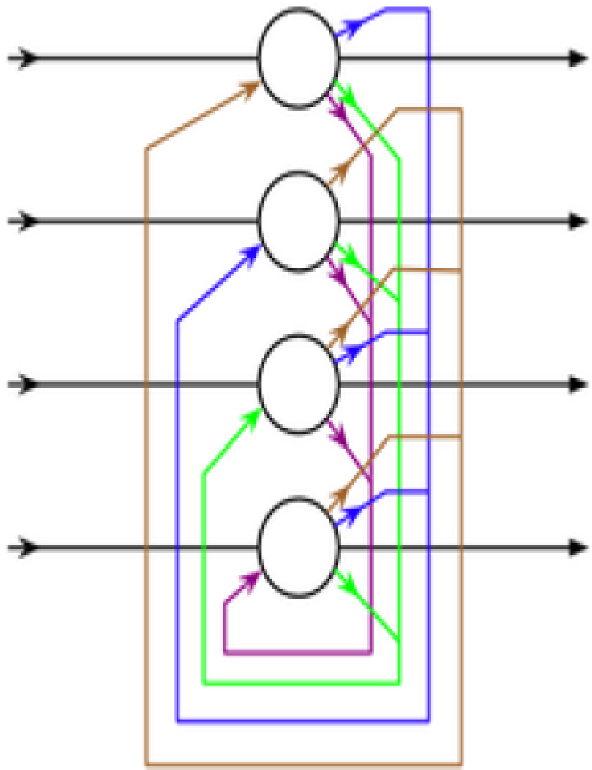
b. What features of biological neural network make it superior to most sophisticated AI computer system? (8)

Answer: Page 213 of Text Book

Q.8 a. What is Hopfield model of a neural network? Explain. (6)

Answer:

A **Hopfield network** is a form of recurrent artificial neural network invented by John Hopfield. Hopfield nets serve as content-addressable memory systems with binary threshold nodes. They are guaranteed to converge to a local minimum, but convergence to a false pattern (wrong local minimum) rather than the stored pattern (expected local minimum) can occur. Hopfield networks also provide a model for understanding human memory.



A Hopfield net with four nodes.

Structure

The units in Hopfield nets are binary threshold units, i.e. the units only take on two different values for their states and the value is determined by whether or not the units' input exceeds their threshold. Hopfield nets normally have units that take on values of 1 or -1, and this convention will be used throughout the article. However, other literature might use units that take values of 0 and 1.

Every two units i and j of a Hopfield network have a connection that is described by the connectivity weight w_{ij}. In this sense, the Hopfield network can be formally described as a complete undirected graph $G = \langle V, f \rangle$, where V is a set of McCulloch-Pitts neurons and $f : V^2 \rightarrow R$ is a function that links pairs of nodes to a real value, the connectivity weight.

The connections in a Hopfield net typically have the following restrictions:

- $w_{ii} = 0, \forall i$ (no unit has a connection with itself)
- $w_{ij} = w_{ji}, \forall i, j$ (connections are symmetric)

The requirement that weights be symmetric is typically used, as it will guarantee that the energy function decreases monotonically while following the activation rules, and the network may exhibit some periodic or chaotic behaviour if non-symmetric weights are used. However, Hopfield found that this chaotic behavior is confined to relatively small parts of the phase space, and does not impair the network's ability to act as a content-addressable associative memory system.

Updating

Updating one unit (node in the graph simulating the artificial neuron) in the Hopfield network is performed using the following rule:

$$s_i \leftarrow \begin{cases} '1' & \text{if } \sum_j w_{ij} s_j > \theta_i, \\ '-1' & \text{otherwise.} \end{cases}$$

where:

- w_{ij} is the strength of the connection weight from unit j to unit i (the weight of the connection).
- s_j is the state of unit j.
- θ_i is the threshold of unit i.

Updates in the Hopfield network can be performed in two different ways:

- **Asynchronous**: Only one unit is updated at a time. This unit can be picked at random, or a pre-defined order can be imposed from the very beginning.
- **Synchronous**: All units are updated at the same time. This requires a central clock to the system in order to maintain synchronization. This method is less realistic, since biological or physical systems lack a global clock that keeps track of time.

Initialization and Running

Initialization of the Hopfield Networks is done by setting the values of the units to the desired start pattern. Repeated updates are then performed until the network converges to an attractor pattern. In the context of Hopfield Networks, an attractor pattern is a pattern that cannot change any value within it under updating.

Training

Training a Hopfield net involves lowering the energy of states that the net should "remember". This allows the net to serve as a content addressable memory system, that is to say, the network will converge to a "remembered" state if it is given only part of the state. The net can be used to recover from a distorted input to the trained state that is most similar to that input. This is called associative memory because it recovers memories on the basis of similarity. For example, if we train a Hopfield net with five units so that the state (1, 0, 1, 0, 1) is an energy minimum, and we give the network the state (1, 0, 0, 0, 1) it will converge to (1, 0, 1, 0, 1). Thus, the network is properly trained when the energy of states which the network should remember are local minima.

Learning Rules

There are various different learning rules that can be used to store information in the memory of the Hopfield Network. It is desirable for a learning rule to have both of the following two properties:

- *Local*: A learning rule is *local* if each weight is updated using information available to neurons on either side of the connection that is associated with that particular weight.
- *Incremental*: New patterns can be learned without using information from the old patterns that have been also used for training. That is, when a new pattern is used for training, the new values for the weights only depend on the old values and on the new pattern.[1]

These properties are desirable, since a learning rule satisfying them is more biologically plausible. For example, since the human brain is always learning new concepts, one can reason that human learning is incremental. A learning system that would not be incremental would generally be trained only once, with a huge batch of training data.

b. What is self-organization network? (6)

Answer:

A **self-organizing Network** (**SON**) is an automation technology designed to make the planning, configuration, management, optimization and healing of mobile radio access networks simpler and faster. SON functionality and behavior has been defined and specified in generally accepted mobile industry recommendations produced by organizations such as 3GPP (3rd Generation Partnership Project) and the NGMN (Next Generation Mobile Networks).

SON has been codified within 3GPP Release 8 and subsequent specifications in a series of standards including 36.902,[1] as well as public white papers outlining use cases from the NGMN.[2] The first technology making use of SON features will be Long Term Evolution (LTE), but the technology has also been retro-fitted to older radio access technologies such as Universal Mobile Telecommunications System (UMTS). The LTE specification inherently supports SON features like Automatic Neighbor Relation (ANR) detection, which is the 3GPP LTE Rel. 8 flagship feature.[3]

Newly added base stations should be self-configured in line with a "plug-and-play" paradigm, while all operational base stations will regularly self-optimize parameters and algorithmic behavior in response to observed network performance and radio conditions. Furthermore, self-healing mechanisms can be triggered to temporarily compensate for a detected equipment outage, while awaiting a more permanent solution.

SON architectural types

Self-organizing networks are commonly divided into three major architectural types.

Distributed SON

In this type of SON (D-SON), functions are distributed among the network elements at the edge of the network, typically the ENodeB elements. This implies a certain degree of localization of functionality, and is normally supplied by the network equipment vendor manufacturing the radio cell.

Centralized SON

In centralized SON (C-SON), function are more typically concentrated closer to higher-order network nodes or the network OSS, to allow a broader overview of more edge elements and coordination of e.g. load across a wide geographic area. Due to the need to inter-work with cells supplied by different equipment vendors, C-SON systems are more typically supplied by 3rd parties like Celcite or Cisco.

Hybrid SON

Hybrid SON is a mix of centralized and distributed SON, combining elements of each in a hybrid solution.

SON sub-functions

Self-organizing network functionalities are commonly divided into three major subfunctional groups, each containing a wide range of decomposed use cases.

Self-configuration functions

Self-configuration strives towards the "plug-and-play" paradigm in the way that new base stations shall automatically be configured and integrated into the network. This means both connectivity establishment, and download of configuration parameters and software. Self configuration is typically supplied as part of the software delivery with each radio cell by equipment vendors. When a new base station is introduced into the network and powered on, it gets immediately recognized and registered by the network. The neighboring base stations then automatically adjust their technical parameters (such as emission power, antenna tilt, etc.) in order to provide the required coverage and capacity, and, in the same time, avoid the interference.

Self-optimization functions

Every base station contains hundreds of configuration parameters that control various aspects of the cell site. Each of these can be altered to change network behavior, based on observations of both the base station itself, and measurements at the mobile station or handset. One of the first SON features establishes neighbor relations automatically (ANR), while others optimize random access parameters or mobility robustness in terms of handover oscillations. A very illustrative use case is the automatic switch-off of a percent of base stations during the night hours. The neighboring base station would then re-configure their parameters in order to keep the entire area covered by signal. In case of a sudden growth in connectivity demand for any reason, the "sleeping" base stations "wake up" almost instantaneously. This mechanism leads to significant energy savings for operators.

Self-healing functions
When some nodes in the network become inoperative, self-healing mechanisms aim at reducing the impacts from the failure, for example by adjusting parameters and algorithms in adjacent cells so that other nodes can support the users that were supported by the failing node. In legacy networks, the failing base stations are at times hard to identify and a significant amount of time and resources is required to fix it. This function of SON permits to spot such a failing base stations immediately in order to take further measures, and ensure no or insignificant degradation of service for the users.

c. Discuss the limitations of Neural Networks. (4)

Answer:
1**. They are black box** - that is the knowledge of its internal working is never known
2. **To fully implement a standard neural network architecture would require lots of computational resources** - for example you might need like 100,000 Processors connected in parallel to fully implement a neural network that would "somewhat" mimic the neural network of a cat's brain - or I may say its a greater computational burden
3. **Remember the No Free Lunch Theorem** - a method good for solving 1 problem might not be as good for solving some other problem - Neural Networks though they behave and mimic the human brain they are still limited to specific problems when applied
4**. Since applying neural network for human**-related problems requires Time to be taken into consideration but its been noted that doing so is hard in neural networks
5. **They are just approximations of a desired solution and errors in them is inevitable**

Q.9 a. Discuss the use of Artificial intelligence techniques in E-Commerce applications. (8)

Answer:
E-Commerce applications present unique challenges and opportunities for developing various data mining, text mining, and web mining techniques for business intelligence and knowledge management purposes. Based on more than one decade of research funded by NSF and other major commercial companies (HP, SAP, DEC, AT&T, Commerce One, etc.), the University of Arizona Artificial Intelligence Lab has conducted ECommerce
Intelligence and Mining research in the following application areas:
 Business Intelligence: Multilingual e-commerce portals and knowledge mapping systems;
 Group Decision Support Systems (GDSS): Text mining and visualization to support collaboration;
 Credit Rating: Data mining for international credit rating analysis;
 ERP Knowledge Management: Knowledge mapping research for ERP content mining;
 Patent Analysis: Text and citation based analysis of international patents;
 E-Commerce Recommender Systems: Graph-based models for customer relation management (CRM) and product recommendation;
 E-Commerce Security Analysis: Text and web mining for detecting fraudulent web

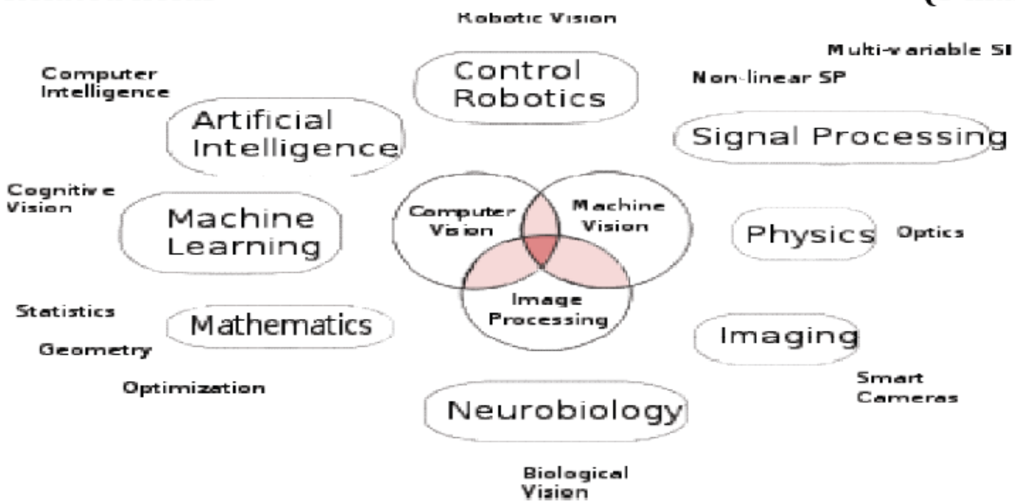
sites and e-commerce contents;
 Stock Prediction Systems: Text mining based stock outbreak detection;
 E-Commerce Marketing and Survey: Opinion mining for Web 2.0 customer-generated contents and sentiment analysis.
 Finance and Accounting Text Mining: Earnings, return, volatility, and volume prediction based on mass media (press) and social media.
Students can also describe:
 Artificial Intelligence in different E-Commerce models like B-2-B, C-2-C, B-2-C etc
 AI in Online auction
 AI in Online negotiation.
 Etc

b. Write a short note on: (4×2)
(i) Computer vision
(ii) Machine perception

Answer:
i) Computer vision is a field that includes methods for acquiring, processing, analyzing, and understanding images and, in general, high-dimensional data from the real world in order to produce numerical or symbolic information, *e.g.*, in the forms of decisions. A theme in the development of this field has been to duplicate the abilities of human vision by electronically perceiving and understanding an image. This image understanding can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and learning theory. Computer

vision has also been described as the enterprise of automating and integrating a wide ran of processes and representations for vision perception.

Related fields **(3 marks)**



Applications for computer vision **(1 marks)**

One of the most prominent application fields is medical computer vision or medical ima processing. This area is characterized by the extraction of information from image data the purpose of making a medical diagnosis of a patient. Generally, image data is in the fo of microscopy images, X-ray images, angiography images, ultrasonic images, a tomography images. An example of information which can be extracted from such ima data is detection of tumours, arteriosclerosis or other malign changes. It can also measurements of organ dimensions, blood flow, etc. This application area also suppo medical research by providing new information, *e.g.*, about the structure of the brain, about the quality of medical treatments. Applications of computer vision in the medical a also includes enhancement of images that are interpreted by humans, for exam ultrasonic images or X-ray images, to reduce the influence of noise.

9.b ii) Machine perception is a term that is used to identify the capability of a compu system to interpret data in a manner that is similar to the way humans use their senses relate to the world around them.[1] The basic method that the computers take in and respo to their environment is through the attached hardware. Until recently input was limited t keyboard, a mouse, or external hard drives, but advances in technology, both in hardware a software, have allowed computers to take in sensory input in a way similar to humans.[1]

Machine perception allows the computer to use this sensory input, as well as conventio computational means of gathering information to gather information with greater accur and to present it in a way that is more comfortable for the user.[1] These include compu vision, machine hearing, and machine touch.

Machine Vision

Computer vision is a field that includes methods for acquiring, processing, analyzing, and understanding images and, in general, high-dimensional data from the real world in order to produce numerical or symbolic information, e.g., in the forms of decisions. Computer vision has many applications already in use today such as facial recognition, geographical modeling, and even aesthetic judgment.

Machine Hearing

Machine hearing is the ability of a computer or machine to take in and process sound data such as music or speech. This area has a wide range of application including music recording and compression, speech synthesis, and speech recognition.[3] Many commonly used devices such as a smartphones, voice translators, and even cars make use of some form of machine hearing.

Machine Touch

Machine touch is an area of machine perception where tactile input from a user or from the environment can be processed by a machine or computer. Applications for this could allow for more direct communication with mechanical devices. Common examples are touch pads
on devices like phones and laptops that allow for a more comfortable interaction with the given device.

TEXT BOOK

Introduction to Artificial Intelligence, Rajendra Akerkar, PHI, 2005