**Abstract**

Cancer is a group of diseases involving abnormal cell growth According to American cancer society, Cancer continues to be the second most common cause of death in the US, after heart disease. A total of 1.9 million new cancer cases and 609,360 deaths from cancer are expected to occur in the US in 2022, which is about 1,670 deaths a day.

no permanent cure has been developed to combat cancer, early detection is crucial for treatment and survival of patient.

Image recognition and deep learning have been used effectively in detection and treatment of several dangerous diseases, helping in early diagnosis and treatment.

The risk of death from cancer dropped by about 2% a year from 2015 through 2019 compared to 1% a year during the 1990s. Accelerating declines in the cancer death rate show the power of prevention, screening, early diagnosis, treatment.

Deep learning can be used to analyze features allowing detection of breast cancer.

Two of the most common imaging used in breast cancer detection are histopathology and mammography.

In our research we aim to compare different CNN architectures (densenet-121 and inception v-4) on each type of the two imaging, while analyzing the results of the different methods for detection and classification.

Key Words: Breast Cancer; Image recognition; Deep Learning; classification; CNN architectures; Inception v-4; densenet-121

**1. Introduction**

Breast cancer is the second most diagnosed cancer worldwide.[1]

Breast cancer occurs in four main types: normal, benign, in-situ carcinoma and invasive carcinoma [2].

In situ carcinoma, the cancer does not affect other organs other than mammary duct lobule system. Benign is not classified as a harmful cancer and involves a minor change in the breast structure. Invasive carcinoma is the deadliest type out of all the four main breast cancer types because it can spread out to all other organs.   
Breast cancer can be diagnosed using one of two approaches: histopathological image analysis or mammography.

Histopathological images are microscopic images of breast tissue that are extremely useful in early treatment of the cancer.  
Mammography is specialized medical imaging that uses a low-dose x-ray system to see inside the breasts. A mammography exam, called a mammogram, aids in the early detection and diagnosis of breast diseases in women.

The main difference between:  
mammography is an earlier type of imaging, before breast tissue is collected for histopathology, an x ray image inside the breasts allows us to search for lumps indicating cancer cells, if there is an indication for breast cancer, breast tissue is collected for analysis under microscope for more accurate diagnosis.

We aim to compare and find which one of the 2 models will satisfy with most accuracy for each type while analyzing the results.

We will use each CNN architectures on mammography and histopathology datasets

And as well study each architecture and make a comparison based on the structure aspects of the models and how each of them compares to the two types of images.

References:

1 S. Germano and L. O'Driscoll, Curr. Cancer Drug Targets, 2009, 9, 398–418.

**2 Background and Related Work**

**2.1 Background –** ANN  
These are computing systems inspired by the biological neural networks in animal's minds.  
An ANN is based on a collection of connected units or nodes called artificial neurons (nerve cell**)**, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron receives signals then processes them and can signal neurons connected to it. The "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold.

Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer) to the last layer (the output layer), possibly after traversing the layers multiple times.  
  
With this the artificial neural network can perform learning through adjustment of the weights after testing the results of the model.

2.1.2 Weights

Each neuron in a neural network computes an output value by applying a specific function to the input values received from the receptive field in the previous layer. The function that is applied to the input values is determined by a vector of weights and a bias (typically real numbers). Learning consists of iteratively adjusting these biases and weights.

2.1.3 Activation Function

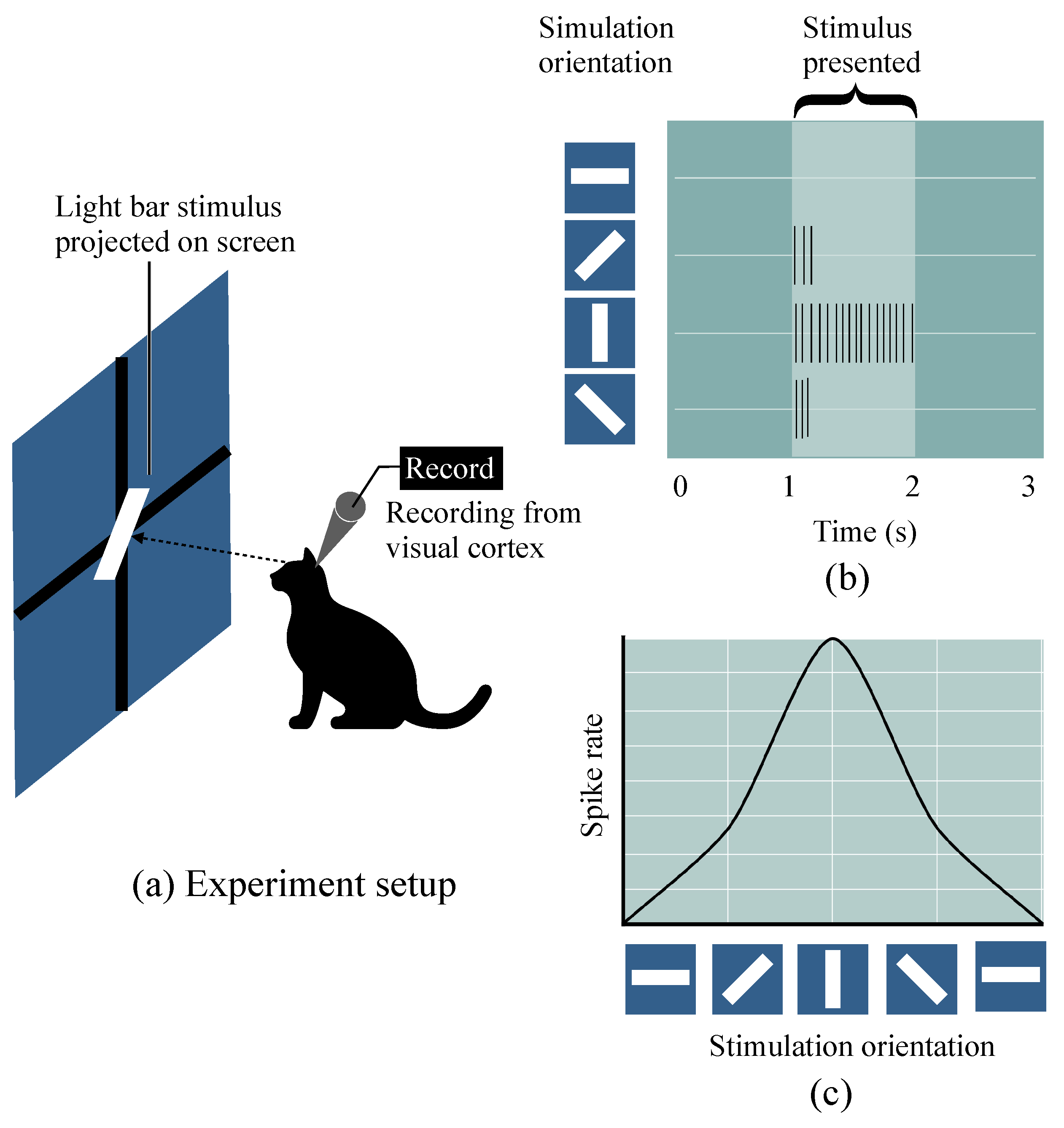
In artificial neural networks, the activation function of a node defines the output of that node given an input or set of inputs.  When compared to a neuron-based model in nature, the activation function output is to be inputted to the next neuron.

Diagram, line chart

Description automatically generated

2.2.1 CNN (Convolutional neural network)  
  
Convolutional neural networks are a specialized type of artificial neural networks that use a mathematical operation called convolution in place of general matrix multiplication in at least one of their layers.[13] They are specifically designed to process pixel data and are used in image recognition and processing.

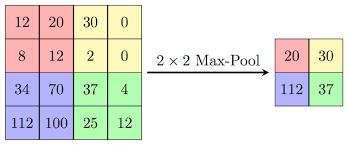
[13] Ian Goodfellow and Yoshua Bengio and Aaron Courville (2016). [*Deep Learning*](https://www.deeplearningbook.org/). MIT Press. p. 326.

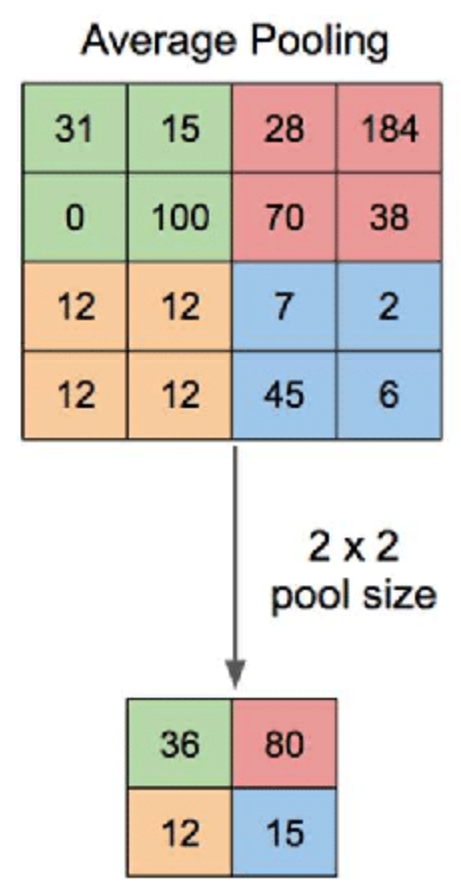
Development of CNN, specifically the type of filters used, was inspired by studies of organization of animal visual cortex such as Hubel & Wiesel.  
  
  
A convolutional neural network consists of an input layer, hidden layers and an output layer. In any feed-forward neural network, any middle layers are called hidden because their inputs and outputs are masked by the activation function and final convolution.  
Digital images are stored as pixel values in a two-dimension array.  
In a CNN model we then go over the data with filters (such as filters that search for a certain color or a type of edge) and extract features from the image, and as layers feed onto each other extracted features can become more complex in deeper layers.

2.2.2 Convolutional layers   
In a CNN, the input is a tensor with a shape: (number of inputs) × (input height) × (input width) × (input channels). After passing through a convolutional layer, the image becomes abstracted to a feature map, also called an activation map, with shape: (number of inputs) × (feature map height) × (feature map width) × (feature map channels).

Convolutional layers convolve the input and pass its result to the next layer. This is like the response of a neuron in the visual cortex to a specific stimulus.

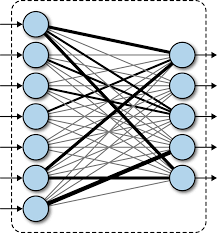
2.2.3 pooling layers  
Pooling layers reduce the dimensions of data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer.  
There are two common types of pooling in popular use: max and average. Max pooling uses the maximum value of each local cluster of neurons in the feature map, while average pooling takes the average value.





2.2.4 Fully Connected Layer

Fully connected layers connect every neuron in one layer to every neuron in another layer. They come as the last layers after convolution and pooling   
and these layers purpose is to take the output of previous layers to classify the image into a label (if for instance we have 3 possible layers: car, dog or cat, the last layer will have 3 nodes, and the weights will point each possibility and strength of the 3 labels).

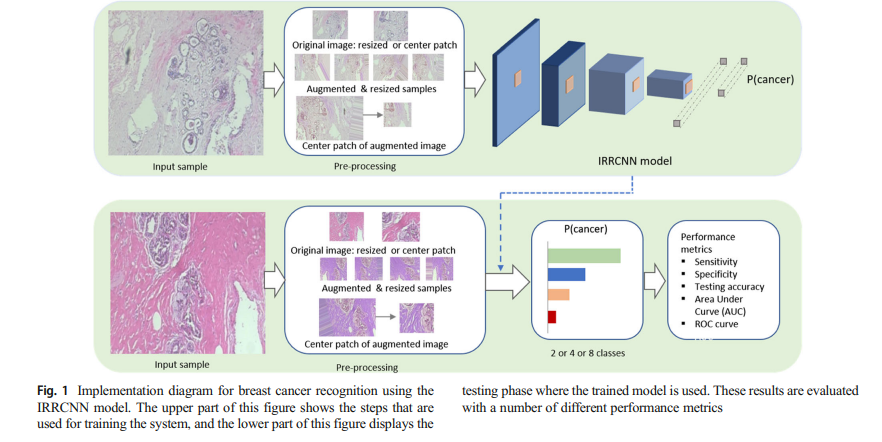


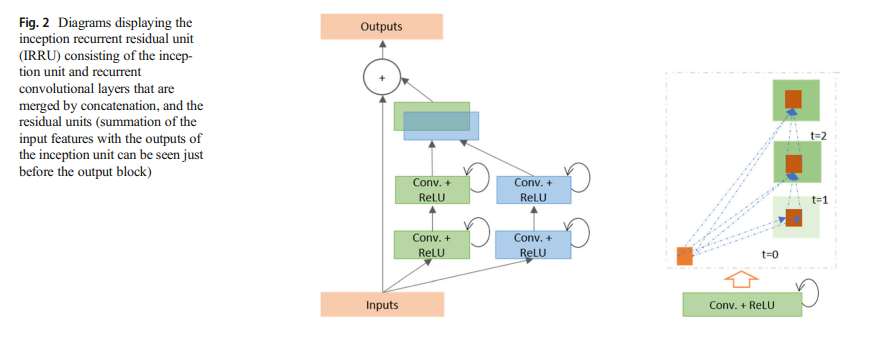
להחליף תמונה

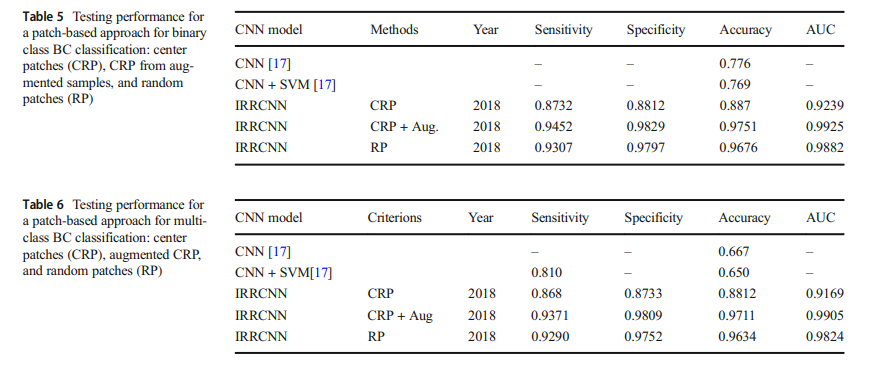
2.3 Inception  
The Inception network comprises of repeating patterns of convolutional design configurations called Inception modules.  
An Inception Module consists of the following components:

* Input layer
* 1x1 convolution layer
* 3x3 convolution layer
* 5x5 convolution layer
* **Max pooling layer**
* **Concatenation layer**

The max-pooling layer and concatenation layer are yet to be introduced within this article. Let’s address this.  
Pooling down-samples the input data to create a smaller output with a reduced height and width.  
Within an Inception module, we add padding(same) to the max-pooling layer to ensure it maintains the height and width as the other outputs (feature maps) of the convolutional layers within the same Inception module.  
By doing this, we ensure we can concatenate the outputs of the max-pooling layer with the outputs of the conv layers within the concatenation layer.

**2.4 Background** Related works  
in [Alom MZ, Yakopcic C, Nasrin MS, Taha TM, Asari VK. Breast Cancer Classification from Histopathological Images with Inception Recurrent Residual Convolutional Neural Network. J Digit Imaging. 2019 Aug;32(4):605-617. doi: 10.1007/s10278-019-00182-7. PMID: 30756265; PMCID: PMC6646497]  
 presents a method for classifying breast cancer using histopathological images and a type of deep learning model called an Inception Recurrent Residual Convolutional Neural Network (IRRCNN).   
IRRCNN is a hybrid DCNN architecture based on inception, residual networks and RCNN model.  
The overall model consists of several convolution layers, IRRUs, transition blocks, and a SoftMax at the output layer.  
the datasets used were Breakhis and BC Challenge Dataset 2015 and variation of datasets with data augmentation.  
  




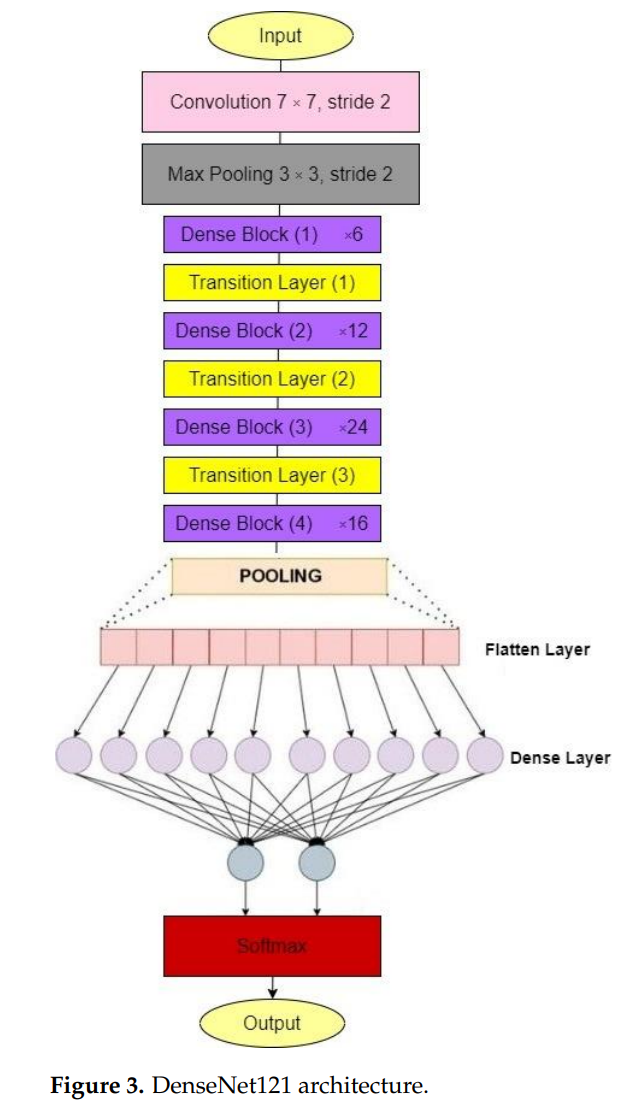


in [Ziliang Zhong et al 2020 J. Phys.: Conf. Ser. 1651 012143] researchers test variants of densenet201 on classification of cell from Kaggle PCAM dataset of histopathological images of cells with labeling of cancer cells.  
the researchers noted the use of densenet201 (TTA) where densenet201 was used for the training set, but the test set is augmented.  
the use of data augmentation can increase the diversity of samples by making minor changes to the images.  
they then compared their results with vgg19 and resnet34 model.

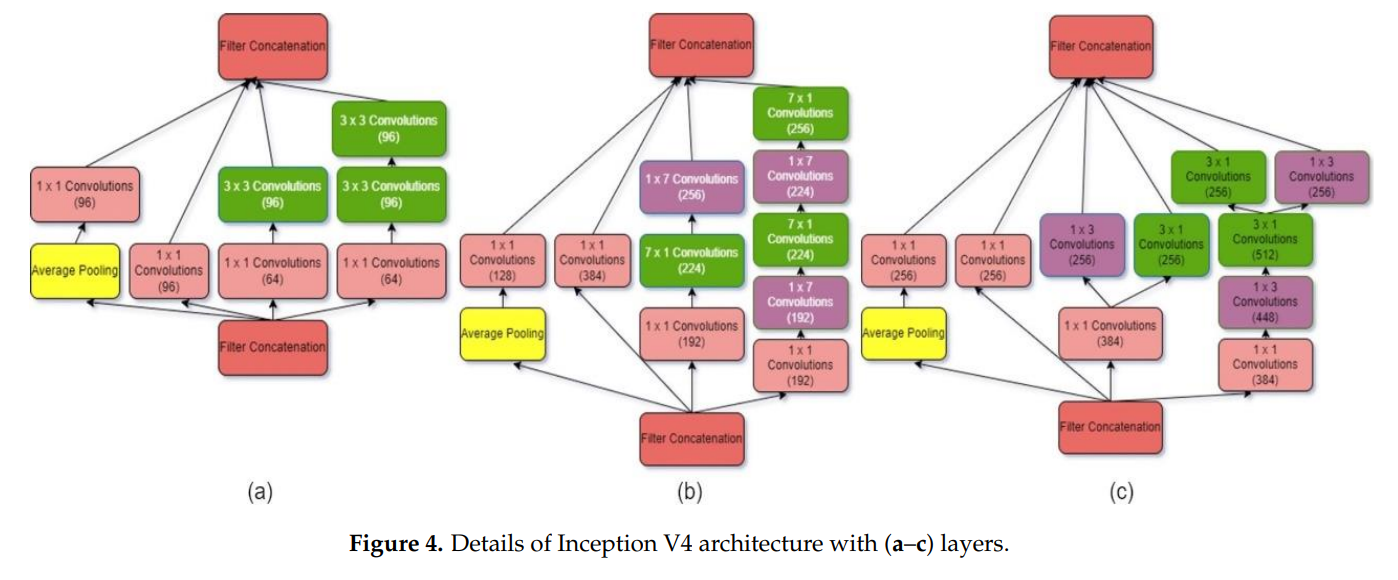
|  |  |
| --- | --- |
| models | accuracy |
| Resnet34 | 0.975 |
| Vgg19 | 0.965 |
| Densenet201 | 0.980 |
| densenet201(TTA) | 0.989 |

In [3] this work the authors employed the Gompertz function to construct a fuzzy ranking algorithm. The benefit of such fusion is that it provides the final prediction for each sample using adaptive weights relay on each classifier confidence scores used to create the ensemble. The Gompertz function was developed on the notion that as an individual aged, mortality reduces exponentially until it approaches an asymptote. It might be useful for fusing the confidence scores of classifiers in a complicated image classification issue, in which the confidence score for a prediction category by a classifier ever achieves absolute zero value but rather some lesser value.

The uses of architectures used in [2] help them to reach the higher accuracy, 2 of the architecture used there were the **DenseNet121** which provides various significantly lowered number of parameters, the reuse of features, and the mitigration of the vanishing gradient.

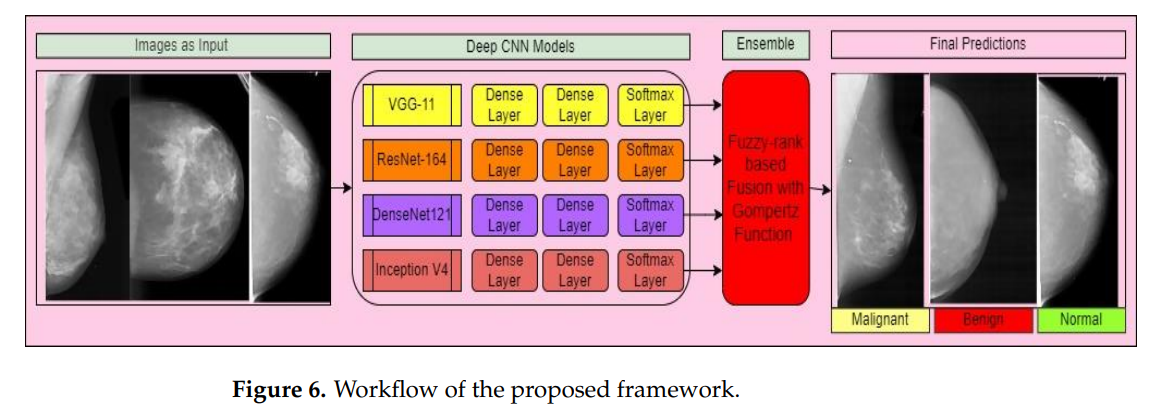


**Inception V4** is a deep CNN architecture that improves on earlier inception family

generations by simplifying the architecture, adding a stem layer, and utilizing more inception modules than Inception v3.

For testing and training they used various of data sets such as: the Breast Cancer Digital Repository (BCDR), the Mini Mammographic Image Analysis Society (Mini-MIAS), INbreast, and the Digital Database for Mammography Screening (DDSM). used an equal number of normal, benign, and malignant mammography images from the whole dataset.

30% for the test set and 70% for the training set used for all the models.

suggested framework for breast cancer classification from mammography images is divided into two stages: producing confidence values from various models and fusing the decision scores utilizing fusion of fuzzy rank and Gompertz function to create final predictions. Figure 6 depicts the workflow of the proposed system.

The findings from the complementary set of classifiers are merged using fuzzy ensemble techniques, which dynamically modify weights to the component deep CNNs depending on the confidence ratings of their predictions. Extensive testing on a range of datasets using a number of measurements reveals the resilience of our method, which frequently surpasses the state-of-the-art in the area. For breast cancer, the suggested framework employed an ensemble model employing the Gompertz function and attained a three-class classification accuracy of 99.32%. It also works well on the overwhelming majority of datasets in the field.

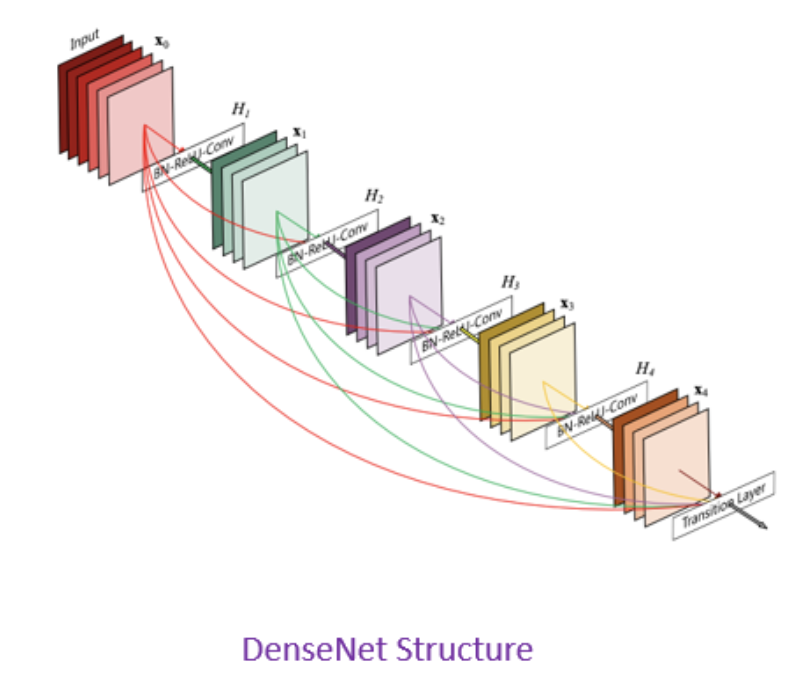
**4.1 Background – DenseNet-121 (Densely Connected Convolutional Networks)**

In a traditional feed-forward Convolutional Neural Network (CNN), each convolutional layer except the first one (which takes in the input), receives the output of the previous convolutional layer and produces an output feature map that is then passed on to the next convolutional layer. Therefore, for 'L' layers, there are 'L' direct connections, one between each layer and the next layer.

However, as the number of layers in the CNN increase, i.e., as they get deeper, the '**vanishing gradient**' problem arises. This means that as the path for information from the input to the output layers increases, it can cause certain information to 'vanish' or get lost which reduces the ability of the network to train effectively.

**4.1.1 Vanishing Gradient problem solve**

DenseNets resolve vanishing gradient problem by modifying the standard CNN architecture and simplifying the connectivity pattern between layers. In a DenseNet architecture, each layer is connected directly with every other layer, hence the name Densely Connected Convolutional Network. For 'L' layers, there are L(L+1)/2 direct connections.



**2.1.2**

**DenseNet Components**

DenseNet components including 4 parts:

* Connectivity
* DenseBlocks
* Growth Rate
* Bottleneck layers

**4.1.2.1 Connectivity**

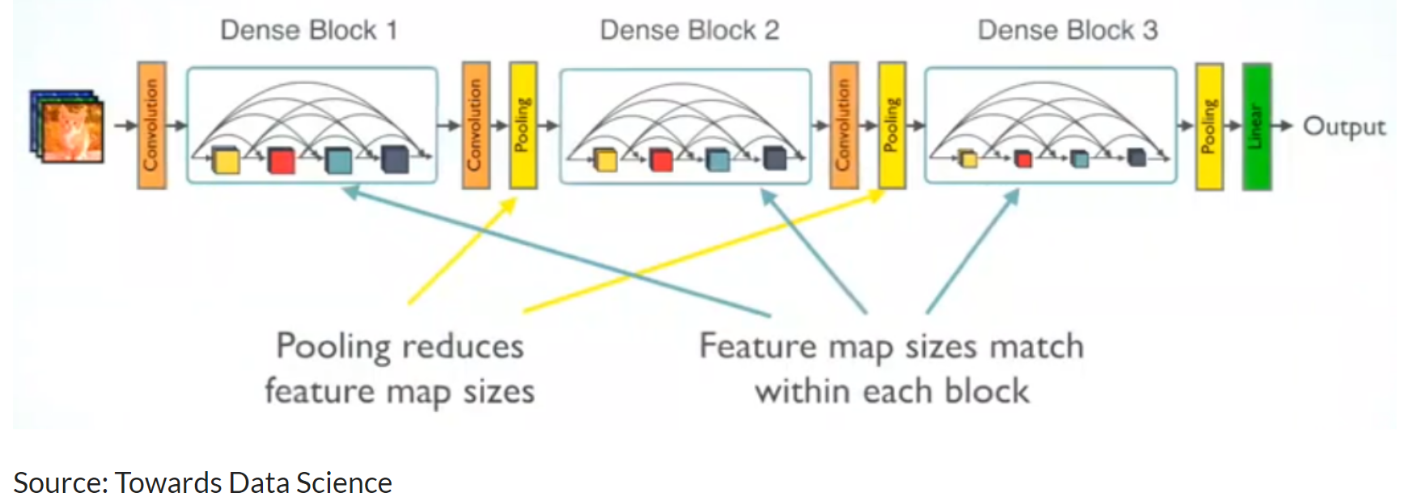
in each layer, the feature maps of all the previous layers are not summed, but concatenated and used as inputs. Consequently, DenseNets require fewer parameters than an equivalent traditional CNN, and this allows for feature reuse as redundant feature maps are discarded. So, the lth layer receives the feature-maps of all preceding layers, x0,...,xl-1, as input:

**4.1.2.2 DenseBlocks**

The use of the concatenation operation is not feasible when the size of feature maps changes. However, an essential part of CNNs is the down-sampling of layers which reduces the size of feature-maps through dimensionality reduction to gain higher computation speeds.

To enable this, DenseNets are divided into DenseBlocks, where the dimensions of the feature maps remains constant within a block, but the number of filters between them is changed. The layers between the blocks are called Transition Layers which reduce the the number of channels to half of that of the existing channels.

For each layer, from the equation above, Hl is defined as a composite function which applies three consecutive operations: batch normalization (BN), a rectified linear unit (ReLU) and a convolution (Conv).



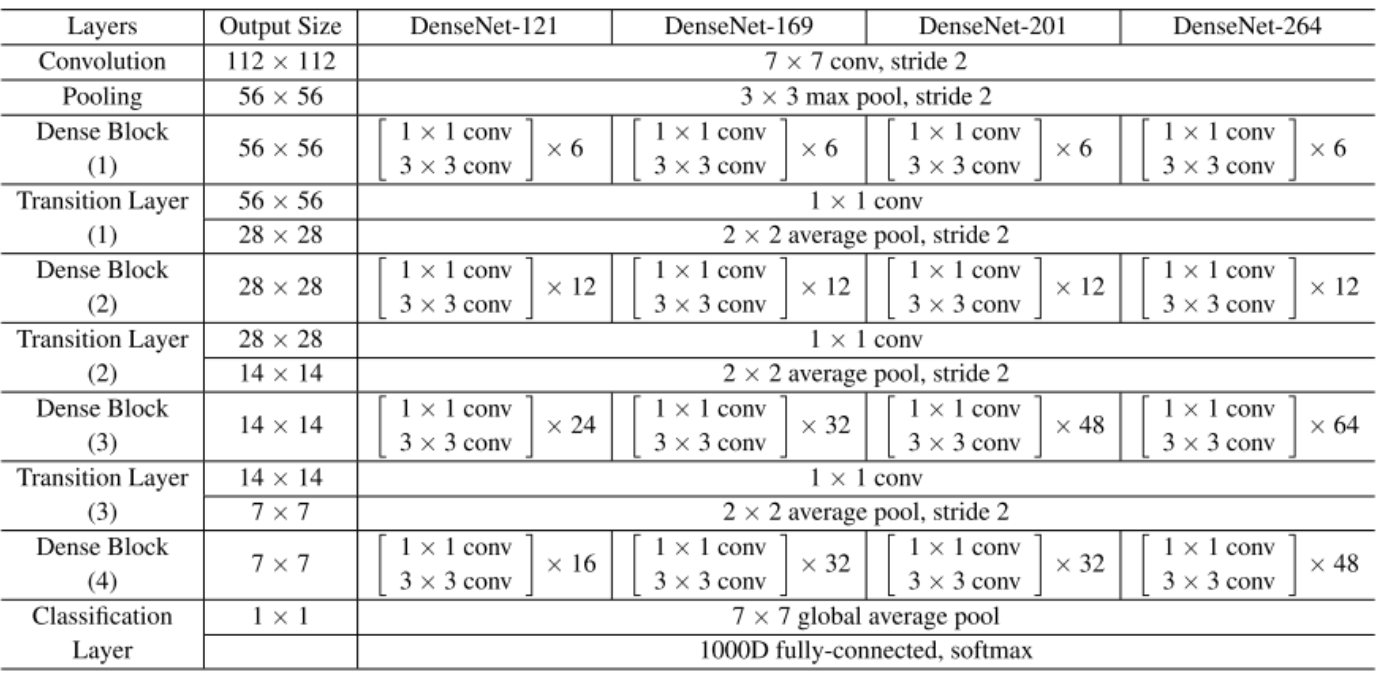
In the above image, a deep DenseNet with three dense blocks is shown. The layers between two adjacent blocks are the transition layers which perform downsampling (i.e. change the size of the feature-maps) via convolution and pooling operations, whilst within the dense block the size of the feature maps is the same to enable feature concatenation.

**4.1.2.3 Growth Rate**

One can think of the features as a global state of the network. The size of the feature map grows after a pass through each dense layer with each layer adding 'K' features on top of the global state (existing features). This parameter 'K' is referred to as the growth rate of the network, which regulates the amount of information added in each layer of the network. If each function H l produces k feature maps, then the lth layer has

**4.1.2.4 Bottleneck layers**

Although each layer only produces k output feature-maps, the number of inputs can be quite high, especially for further layers. Thus, a 1x1 convolution layer can be introduced as a bottleneck layer before each 3x3 convolution to improve the efficiency and speed of computations.

**4.1.3 DenseNet Architecture**

A summarization of the various architectures implemented for the ImageNet database have been provided in the table above. Stride is the number of pixels shifts over the input matrix. A stride of 'n' (default value being 1), indicates that the filters are moved 'n' pixels at a time.

Using the DenseNet-121 architecture to understand the table, we can see that every dense block has varying number of layers (repetitions) featuring two convolutions each; a 1x1 sized kernel as the bottleneck layer and 3x3 kernel to perform the convolution operation.

Also, each transition layer has a 1x1 convolutional layer and a 2x2 average pooling layer with a stride of 2. Thus, the layers present are as follows:

1. Basic convolution layer with 64 filters of size 7X7 and a stride of 2.
2. Basic pooling layer with 3x3 max pooling and a stride of 2.
3. Dense Block 1 with 2 convolutions repeated 6 times.
4. Transition layer 1 (1 Conv + 1 AvgPool).
5. Dense Block 2 with 2 convolutions repeated 12 times.
6. Transition layer 2 (1 Conv + 1 AvgPool).
7. Dense Block 3 with 2 convolutions repeated 24 times.
8. Transition layer 3 (1 Conv + 1 AvgPool).
9. Dense Block 4 with 2 convolutions repeated 16 times.
10. Global Average Pooling layer- accepts all the feature maps of the network to perform classification.
11. Output layer.

Therefore, DenseNet-121 has the following layers:

* 1 7x7 Convolution
* 58 3x3 Convolution
* 61 1x1 Convolution
* 4 AvgPool
* 1 Fully Connected Layer

To summarize DenseNet-121 has 120 Convolutions and 4 AvgPool.

All layers i.e., those within the same dense block and transition layers, spread their weights over multiple inputs which allows deeper layers to use features extracted early on.

**4.1.4 Advantages of the DenseNet**

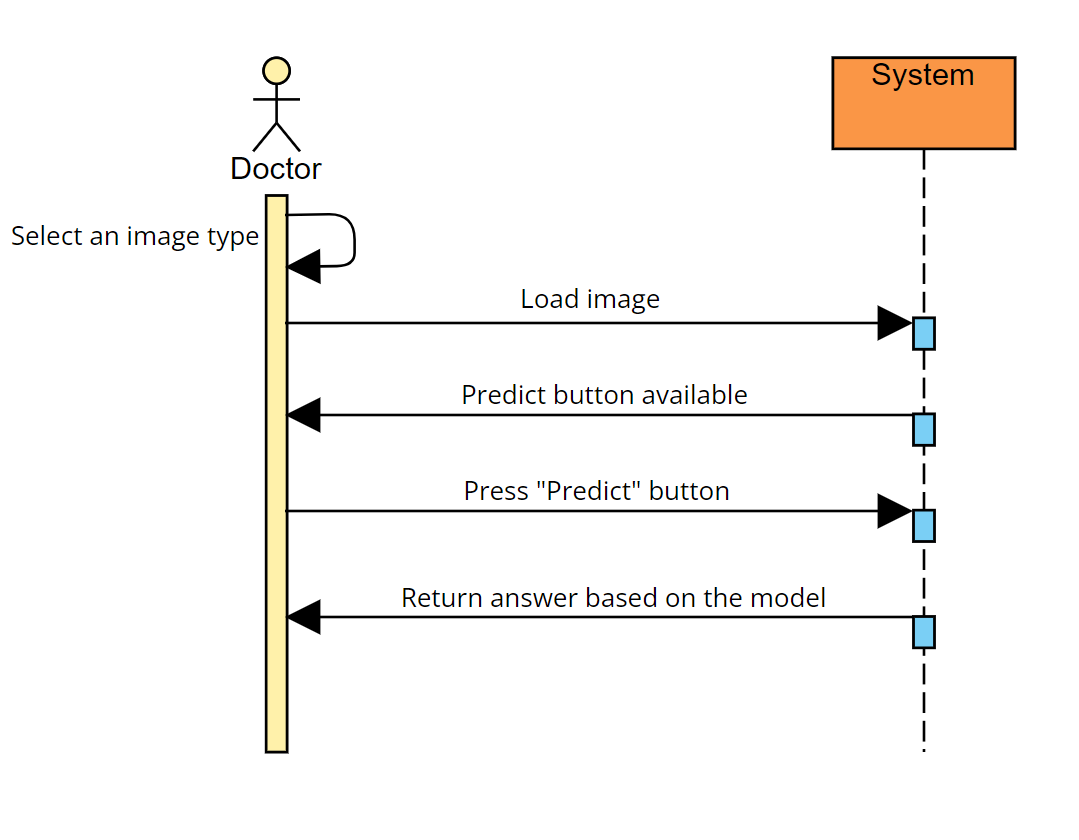
Two of the most obvious Advantages of the DenseNet are

**Parameter efficiency** and **Implicit deep supervision** which result in more compact models and have achieved state of the art of performances and better results across competitive datasets, as compared to their standard CNN counterpart.

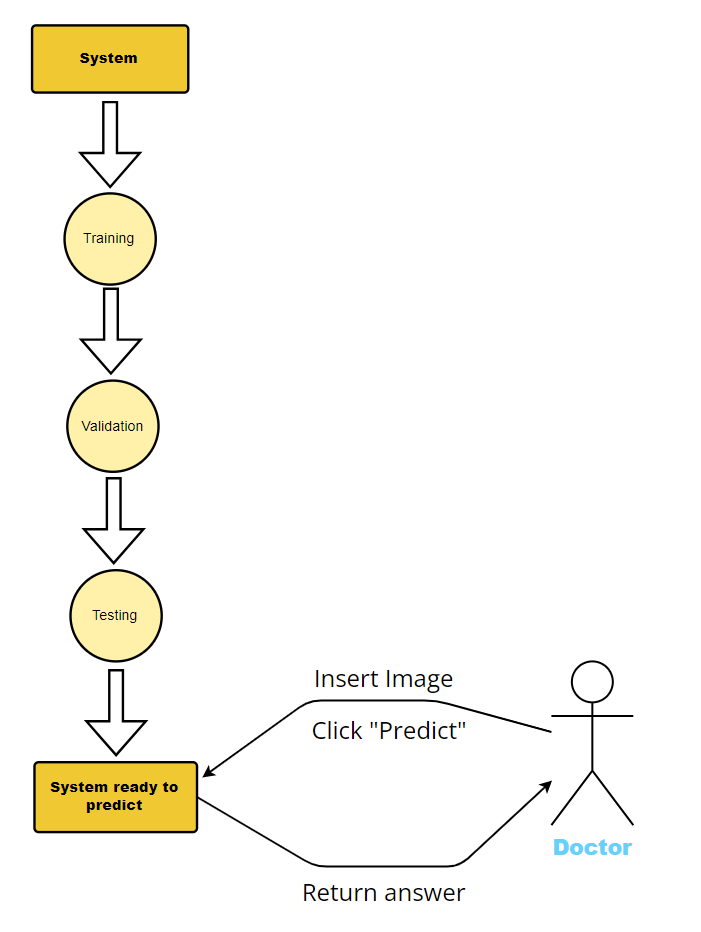
* **Parameter efficiency** – Every layer adds only a limited number of parameters- for e.g. only about 12 kernels are learned per layer
* **Implicit deep supervision** – Improved flow of gradient through the network- Feature maps in all layers have direct access to the loss function and its gradient.

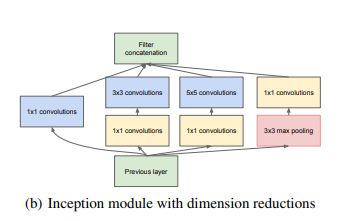
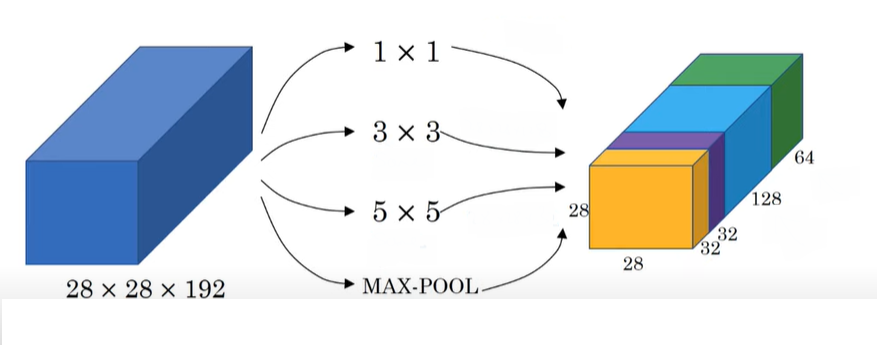
**4.1 Product**

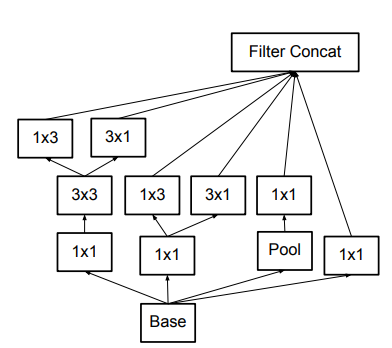
Sequence diagram:

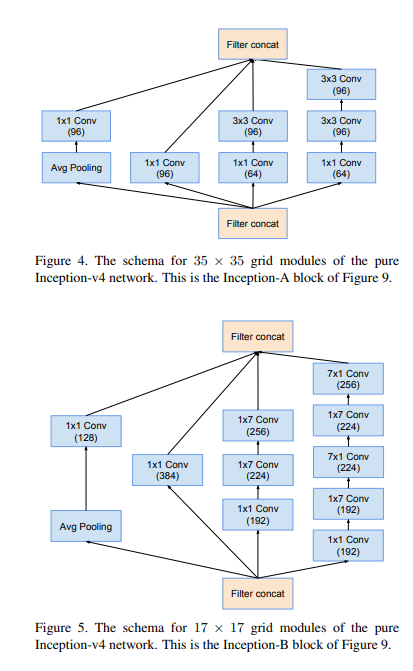
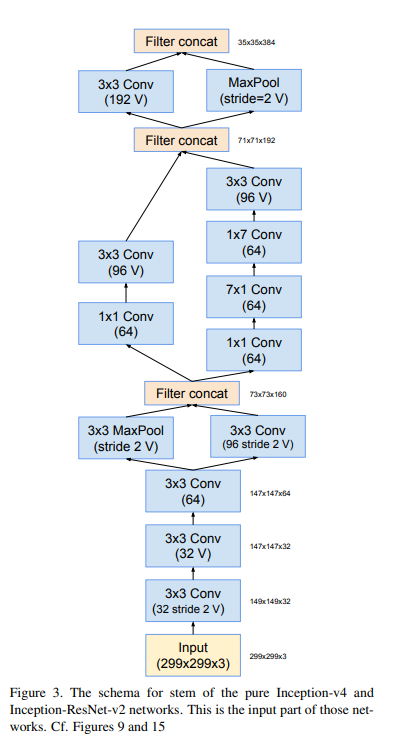


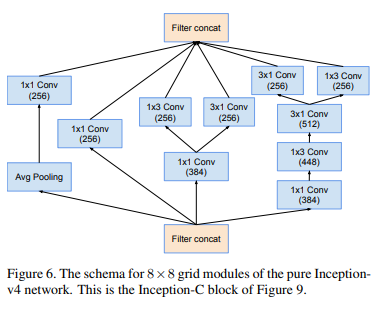
Flowchart diagram:

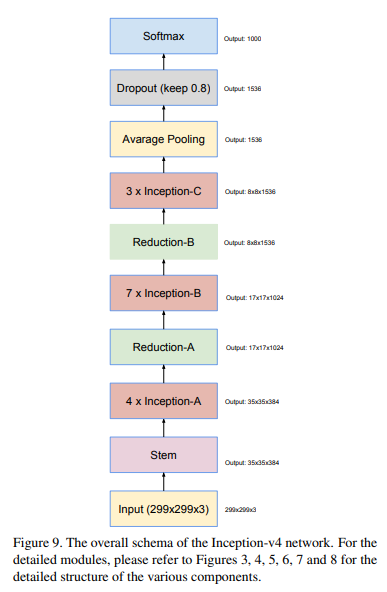


4.2 inception  
the inception network which was introduced in Szegedy et al. 2015a takes a new approach to deep layers models where before different types of convolutions and pooling layers were stacked upon each other but now in an inception module we combine within the same layer different convolutions and a pooling operation which all produces at the layer level an output with same dimensions only different depths which are than concatenated as an output of the layer and passed on unto the next.  
  
inception was made with a goal to reduce parameters needed for the model and at its publication was able to produce better results than former models with much less parameters with 1x1 convolutions.

inception v2 introduced new modules variations and the use of auxiliary classifiers  


Inception v3 is also introduced in the same paper with v2 and is a variation of v2 with batch normalization on the auxiliary classifiers and not just on convolutions.  
the Inception architecture is highly tunable, meaning that there are a lot of possible changes to the number of filters in the various layers that do not affect the quality of the fully trained network.  
inception v4 is an updated version of inception which made use of advancements and newer tools from when v3 was published.  
researchers note not simplifying earlier choices in the architecture resulted in networks that looked more complicated than they needed to be and made changes for version 4.  






4.3 dataset  
machine learning dataset is a collection of data that is used to train the model. A dataset acts as an example to teach the machine learning algorithm how to make predictions.  
In our research we will use two different datasets:  
one for mammography and one for histopathology

4.3.1: mammography dataset:   
RSNA Screening mammography breast cancer detection dataset as found Kaggle.  
this is a recent and up to date dataset.  
included: 54713 files, size of 314.72gb type of dcm,csv  
roughly 8,000 patients in the hidden test set. There are usually but not always 4 images per patient. Note that many of the images use the jpeg 2000 format.

4.3.2 histopathology dataset: