1.

a) Four-fold rotational symmetry: R_0=Rotation of 0^0, R_{90}=Rotation of 90^0, R_{180}=Rotation of 180^0, R_{270}=Rotation of 270^0.

b) I stand for Identity

o	I(R_0)	R_{90}	R_{180}	R_{270}
I(R_0)	I(R_0)	R_{90}	R_{180}	R_{270}
R_{90}	R_{90}	R_{180}	R_{270}	I(R_0)
R_{180}	R_{180}	R_{270}	I(R_0)	R_{90}
R_{270}	R_{270}	I(R_0)	R_{90}	R_{180}

If a , b, c are symmetries of our starfish

- Identity – rotation of 0^0
- Associative – a o (b o c) = (a o b) o c
- Inverse – rotation through the same number of degrees in the opposite direction

The main difference is square having reflection symmetry, but starfish does not have. In formal, is the equilateral is invariant after flipping.

2. The Discrete Fourier Transform (DFT) is the projection of our data (in vector form) onto the harmonic function. Since the root of the unity w is fixed and its size is N, the harmonics of the Fourier matrix F can be predefined. Therefore, we only need to calculate the inner product with our data/signal x. Thus, the DFT X = X_k of the signal x = x_n can be defined as

$$\mathrm{X_k} = \sum_{n=0}^{N-1} x_n * w^{kn}$$

where the N_{th} root of unity is defined as

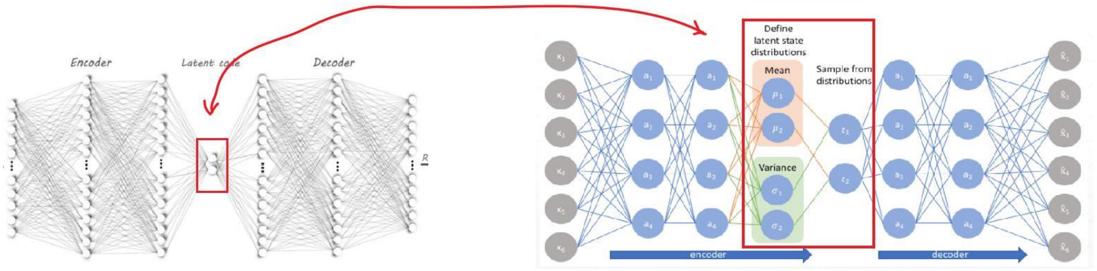
$$\mathrm{w} = \mathrm{e}^{2\pi\mathrm{i}/\mathrm{N}}$$

The Fourier domain has many advantages, including filtering and convolution. We can filter the high frequency part of the image or filter the low frequency part of the image to get the edge of the image. The obtained Fourier space is a one-stop service for filtering and preprocessing operations (such as smoothing, down-sampling, and edge detection).

DFT can reveal hidden harmonic signals, especially those caused by physical systems, such as sound or motion. In fact, DFT is an ideal transformation for signals or data from physical sources, because most physical systems have some oscillations behind them. DFT can calculate the frequency spectrum of the signal. This is a direct inspection information encoded in the frequency, phase and amplitude of the component sinusoidal signal. For example, human speech and hearing use this type of coded signal.

3.

i. Network structure as right image



AE

VAE

ii. AE: ‘theta’ and ‘phi’ which are the parameters that define the encoder and the decoder. the encoder is represented by Gϕ, while the decoder is represented by Fθ and they simply mean the weights and bias of the neural network. So, in the equation, we are summing up the difference between the original image, x`, and the reconstructed image Fθ(gϕ(x`)).

$$L(\theta, \varphi) = \frac{1}{n}\sum_{i=1}^{n} (x^i \ - f_\theta(g_\varphi(x^i)))^2$$

1. Variational Autoencoder

$$-L^{(a)}(\theta, \emptyset; X) = \underbrace{\int_Z q_\emptyset(Z|X) \times \log(P_\theta(X|Z))}_{\text{Reconstruction loss}} - \underbrace{D_{KL}(q_\emptyset(Z|X)||P_\theta(Z))}_{\text{Regularization loss}}$$

$$= \underbrace{-\tfrac{1}{M}\textstyle\sum_{i=1}^{M}\sum_{j=1}^{D}\{X_{i,j}\log P_{i,j} + (1 - X_{i,j})log(1 - P_{i,j})\}}_{\text{Reconstruction loss}} + \underbrace{\tfrac{1}{L}\textstyle\sum_{i=1}^{D}[\tfrac{1}{2}\sum_{j=1}^{J}\{1 \ - \log(\sigma_{i,j}^2) + \ \mu_{i,j}^2 + \sigma_{i,j}^2\}]}_{\text{Regularization loss}}$$

iii. An autoencoder is an artificial neural network that is used to learn effective data encoding in an unsupervised manner. The purpose of an autoencoder is to learn the representation (encoding) of a set of data, usually for dimensionality reduction, and to ignore signal "noise" by training the network.

Like Auto-encoders, the objective of a Variational Auto-encoder is to reconstruct the input. The only difference is that AEs have direct links between encoder and decoder parts, but VAEs have a sampling layer which samples form a distribution (usually a Gaussian) and then feeds the generated samples to the decoder part.

iv. There are three components in autoencoder. They are encoder, decoder and code.

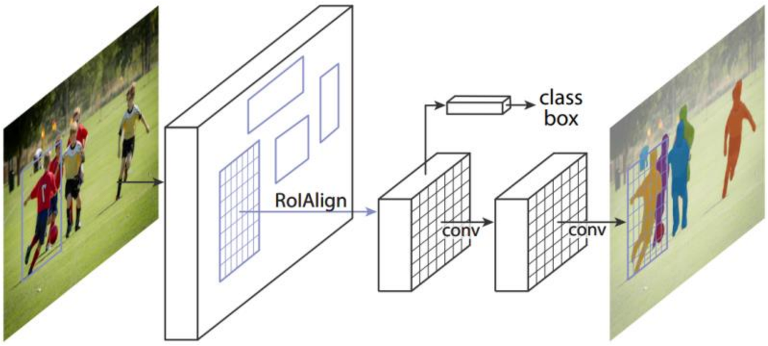
- Encoder: The model learns how to reduce the input dimension and compress the input data into an encoded representation, and VAE samples from latent distribution.
- code: This is the layer containing the compressed representation of the input data. This is the lowest possible dimension of the input data.
- Decoder: The model learns how to reconstruct the encoded data to make it as close to the original input as possible.
- Reconstruction loss: This is a method to measure the performance of the decoder and how close the output is to the original input.

v. Perform Dimensionality reduction by training the network to ignore signal “noise”. Representing data in a lower-dimensional space can improve performance on different tasks, such as classification.

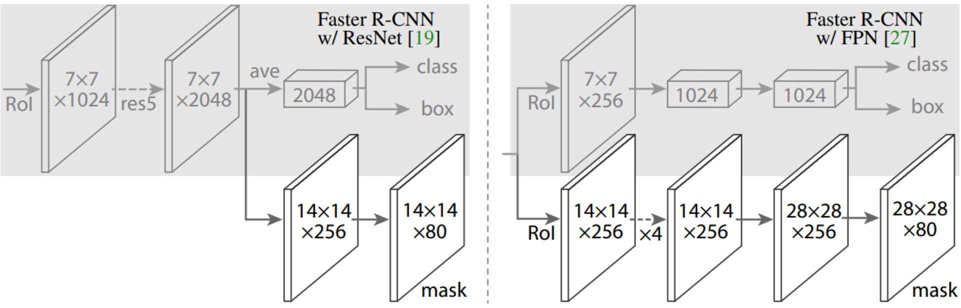
4.

i. He, K., Gkioxari, G., Dollar, P. and Girshick, R., 2017. *Mask R-CNN*. [online] Arxiv.org. Available at: <https://arxiv.org/pdf/1703.06870v3.pdf> [Accessed 13 November 2020].

ii. The Mask R-CNN framework for instance segmentation:



iii. The mask R-CNN expands the faster R-CNN and adds a branch for predicting object masks parallel to the existing branch for bounding box recognition.



The mask R-CNN is conceptually simple: the faster R-CNN has two outputs for each candidate object, a class label and a bounding box offset; for this, we added a third branch, used to output the object mask. Therefore, Mask R-CNN is a natural and intuitive idea. But the additional mask output is different from the class and box output, it needs to extract a finer spatial layout of an object. Next, we will introduce the key elements of Mask R-CNN, including pixel-to-pixel alignment, which is the main missing part of Fast/Faster R-CNN.

iv. Mask R-CNN follows the spirit of Fast R-CNN, which simultaneously applies bounding box classification and regression. In the training process, it defines the multi-task loss on each sampled RoI as $L = L_{cls} + L_{box} + L_{mask}$. The classification loss L_{cls} and the bounding box loss L_{box} are the same. The mask branch has a square kilometer dimensional output for each RoI, and it encodes K binary masks with a resolution of m × m, one for each K category. To this end, we apply a pixel-by-pixel sigmoid and define L_{mask} as the average binary cross-entropy loss. For a RoI related to ground truth class k, L_{mask} is only defined on the kth mask. Experiments show that this formula is the key to obtain good instance segmentation results.

v. Mask R-CNN is simple to train can efficiently detect objects in an image while simultaneously generating a high-quality segmentation mask for each instance;
Mask R-CNN is also fast to train. Training with ResNet-50-FPN on COCO trainval35k takes 32 hours in our synchronized 8-GPU implementation (0.72s per 16-image mini-batch), and 44 hours with ResNet-101-FPN. And it's easy to generalize to other tasks.

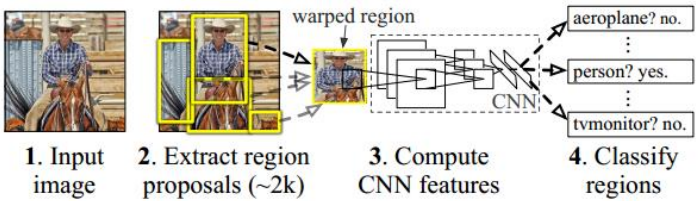
vi. Although the mask R-CNN is fast, we noticed that its design is not optimized for speed and can achieve a better speed/accuracy trade-off, for example, by changing the image size and number of recommendations.

vii. This network can be easily extended to human pose estimation. The position of the key points is modeled as a single thermal mask, and the mask R-CNN is used to predict K masks, and each mask corresponds to K key point types (such as left shoulder, right elbow). This task helps to demonstrate the flexibility of the masked R-CNN. It can be noted that this system utilizes the minimum domain knowledge of human pose, because the experiment is mainly to prove the generality of the mask R-CNN framework. We expect that domain knowledge (for example, modeling structure) will be a supplement to our simple method.
Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In CVPR, 2017.

5. Method: Object detection – combines image classification (predicting the class of one object in an image) and object localization (identifying the location of one or more objects in an image and drawing bounding box around their extent), which localizes and classifies one or more objects in an image.

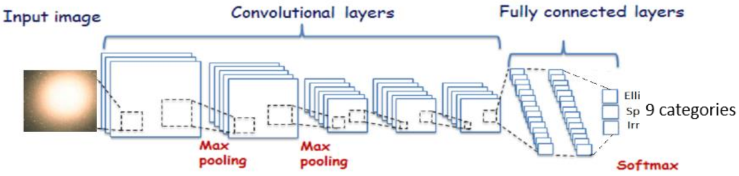
i. To automatically locate all the galaxies in the large images as well as classifying them, I will be using a *Region Based Convolutional Neural Networks (R-CNN)*:

1. The method takes an image as input and extracts region proposals from image(Step 2);
2. Each region proposal is then warped (reshaped) to a fixed size to be passed on as an input to a CNN which is trained by individual galaxy images;
3. The CNN extracts a fixed-length feature vector for each region proposal(Step 3);
4. These features are used to classify region proposals to different category(Step 4);
5. The bounding boxes are refined using bounding box regression so that the object is properly captured by the box.



Input: Image of a galaxy of any one of the nine categories of galaxies.

The Architecture Diagram of Workflow of Image classification using CNN:



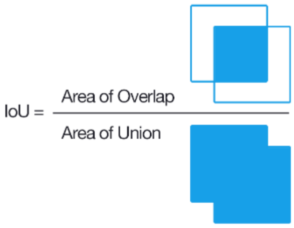
Output: The correct class to which the image of galaxy given as input belongs to.

ii. The CNN can be multi-layer with 3-4 hidden layers and 9 classes or categories with Relu (Rectified Linear Unit) activation function. The loss function used will be adam optimizer (best in general) and categorial cross entropy as we have 9 categories of galaxies. A moderate batch size, combined with a decaying learning rate, is generally used in practice.

iii. Manually check the galaxy boundary box in the large images, since the total large images is only 10, there is not enough sample to make R-CNN region proposals accurate.

iv. The 100 large images will be used as testing set. We then split the rest 4000+1000 images into training and validating set by ratio of 0.85:0.15, the reason behind this decision is our whole dataset is size of 5000, which 85% is enough to train the model. Note the ratio will apply on the galaxy and surface-based images separately.

v. Evaluation Measures: The test validation set is prepared before testing on a new image and when CNN predicts the class images belong to, we count the number of images correctly classified and divide by total number of images to get the *accuracy* value. To evaluate the performance of object detection, we can also use *Intersection-Over-Union(IoU)*: How well the bounding box can locate the object in the image. In other words, how close the predicted bounding box is to the ground truth and Whether the bounding box is classifying the enclosed object correctly.


IoU =
Area of Overlap
Area of Union