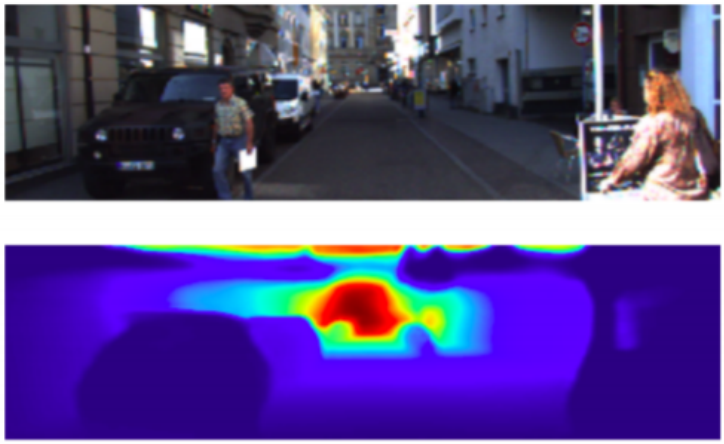
Problem Selection:
The inspiration for selection of the below problem statement is the literature review done during the paper implementation task.
The research problem chosen by me is "**Monocular Depth Estimation**." This is a technique used for estimating the depth of each pixel using an RGB image from a monocular camera. The problem holds a lot of importance in the areas of object detection and varied computer vision tasks like indoor localization and SLAM (Simultaneous Localization and Mapping).

Step 1:
Finding a phenomenon and a question to ask about it

Depth calculation for images is mainly done using depth cameras. They give an image with 4 channels - RGB-D (D stands for depth) rather than a 3 channel image given by monocular cameras. Depth cameras can be StereoCameras which use 2 lenses and use 2 images to calculate the depth of each pixel in the frame of view. Other depth cameras use a normal color camera for RGB channel information and an IR/LiDAR sensor for the depth channel.
These cameras however are very expensive and there is a need for a technique that can use normal 3 channel images for depth calculation. Thus, the underlying question is "**HOW** to estimate the depth of each pixel in an RGB image with dimensions H x W x 3."



RGB image above and estimated depth map below

Step 2:
Understanding the state of the art

Various approaches have been taken towards this research problem. This paper discusses several methods used for monocular depth estimation. Supervised learning methods can be of type classification or regression. **Regression**-based methods directly model the hidden mapping function between RGB images and corresponding depth maps. A **classification** approach groups the depth values into many categories and trains a CNN to determine which bins the pixels fall into. To obtain the continuous depth values, a post-processing step is conducted based on probability scores. (Conditional Random Fields are used after semantic

segmentation). This report will focus on the regression approach. Accuracies ranging from 94-98% have been achieved using the regression based approaches. State-of-the-art RGB-based depth estimation methods use deep learning based methods to train a convolution neural network using large-scale datasets[ref].

Step 3:
Determining the basic ingredients

For a regression based supervised learning approach the input to the model is an RGB image i.e. H x W x 3 and the output is H x W x 1, where depth of each pixel is stored.

Step 4:
Formulating mathematically defined hypothesis

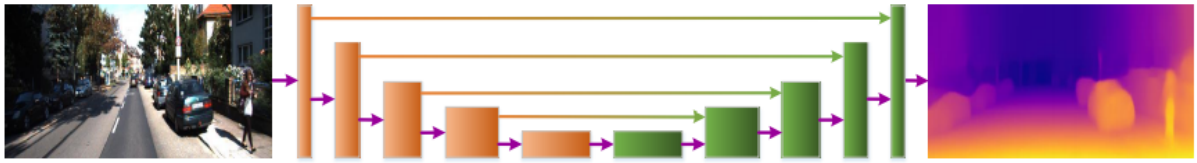
The hypothesis is a mapping from the input image to the depth values which is formed by introducing various intermediate/hidden layers in a deep learning based approach. Complex features of the image are first extracted using the initial layers and are used to calculate the depth values by the further layers.

Step 5:
Selecting the toolkit

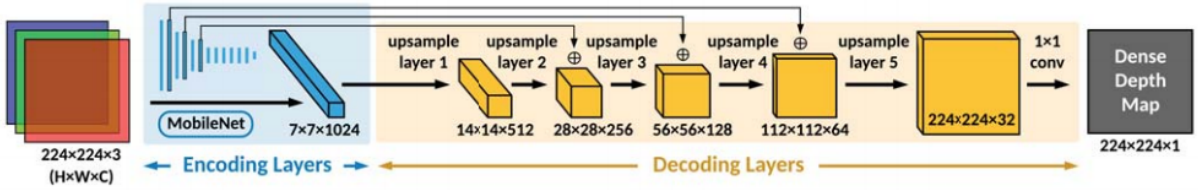
Pytorch is used for implementation of most depth estimation papers.

Step 6:
Planning the model

The architecture uses a fully convolutional encoder-decoder architecture. The encoder extracts high level low-resolution features from the input image. These features are then fed into the decoder, where they are gradually upsampled, refined, and merged to form the final high-resolution output depth map. The encoding layers reduce the dimensions and increase the channels and the decoding layers reduce the channels and increase the dimensions. Various methods can be used for decoders like TransConvolution operation as used in U-Net (for semantic segmentation). There are skip connections from the encoding layers to the decoding layers so that vital information from the earlier layers is not lost.



The orange boxes depict the encoding layers and the green boxes represent the decoding layers



Example architecture of FastDepth that uses MobileNetv1 as the encoder

Step 7:

Implementing, testing and evaluating the model

For supervised learning the datasets that can be used for training are KITTI, NYU Depth, CityScapes and Make3D. The following evaluation metrics are used in depth estimation:

$$\textbf{RMSE} = \sqrt{\frac{1}{|N|}\sum_{i\in N} \| d_i - d_i^* \|^2},$$

$$\textbf{RMSE log} = \sqrt{\frac{1}{|N|}\sum_{i\in N} \| \log(d_i) - \log(d_i^*) \|^2},$$

$$\textbf{Abs Rel} = \frac{1}{|N|}\sum_{i\in N} \frac{|d_i - d_i^*|}{d_i^*},$$

$$\textbf{Sq Rel} = \frac{1}{|N|}\sum_{i\in N} \frac{\|d_i - d_i^*\|^2}{d_i^*},$$

$$\textbf{Accuracies: } \% \text{ of } d_i \text{ s.t. } \max\left(\frac{d_i}{d_i^*}, \frac{d_i^*}{d_i}\right) = \delta < thr,$$

di is the predicted depth value of pixel i, and di∗ stands for the ground truth of depth. N denotes the total number of pixels with real-depth values, and *thr* denotes the threshold

One can use various optimization algorithms for training and compare the performance. Many papers I went through have used RMSprop (uses EMWA). Loss function used is mostly RMSE.

References and Bibliography:

1. Monocular Depth Estimation Using Multi Scale Neural Network And Feature Fusion https://arxiv.org/pdf/2009.09934.pdf

2. Monocular Depth Estimation Based On Deep Learning: An Overview https://arxiv.org/abs/2003.06620

3. Deep Learning based Monocular Depth Prediction: Datasets, Methods and Applications https://arxiv.org/pdf/2011.04123

4. Towards Good Practice for CNN-Based Monocular Depth Estimation https://openaccess.thecvf.com/content_WACV_2020/papers/Fang_Towards_Good_Practice_for_CNN-Based_Monocular_Depth_Estimation_WACV_2020_paper.pdf

5. FastDepth: Fast Monocular Depth Estimation on Embedded Systems

https://arxiv.org/abs/1903.03273