Batch Whitening

Batch Whitening

- Given B samples of a random vector $X \in \mathbb{R}^d$ with covariance Σ_X , generate $Y \in \mathbb{R}^d$ s.t. $\Sigma_Y = I_d$
- Common methods:
  - ZCA whitening
  - Choleski Decomposition
  - Eigen decomposition
-

Using Choleski

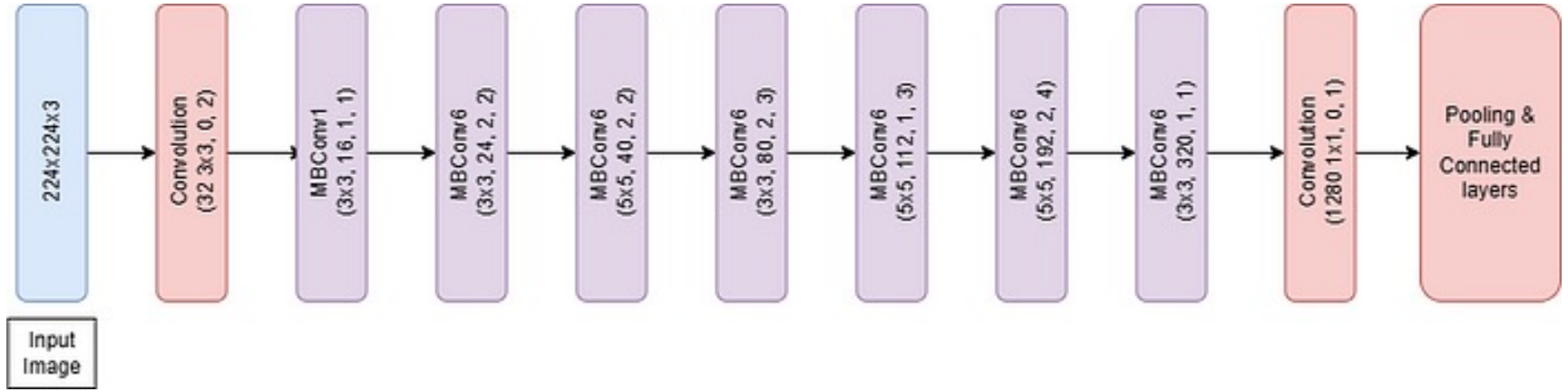
- Compute $\Sigma_x = \frac{1}{B-1} \sum_{i=1}^{B} (x_i - \bar{x})(x_i - \bar{x})^T$
- Using Choleski, find L s.t. $\Sigma_X = L \cdot L^T$
  - L is lower triangular
- Compute L^{-1}
- Whiten by: $y_i = L^{-1}(x_i - \bar{x})^T$

Layer implementation

- Adding γ, β
  - Start with $\beta = 0$ – split from the batch whitening itself

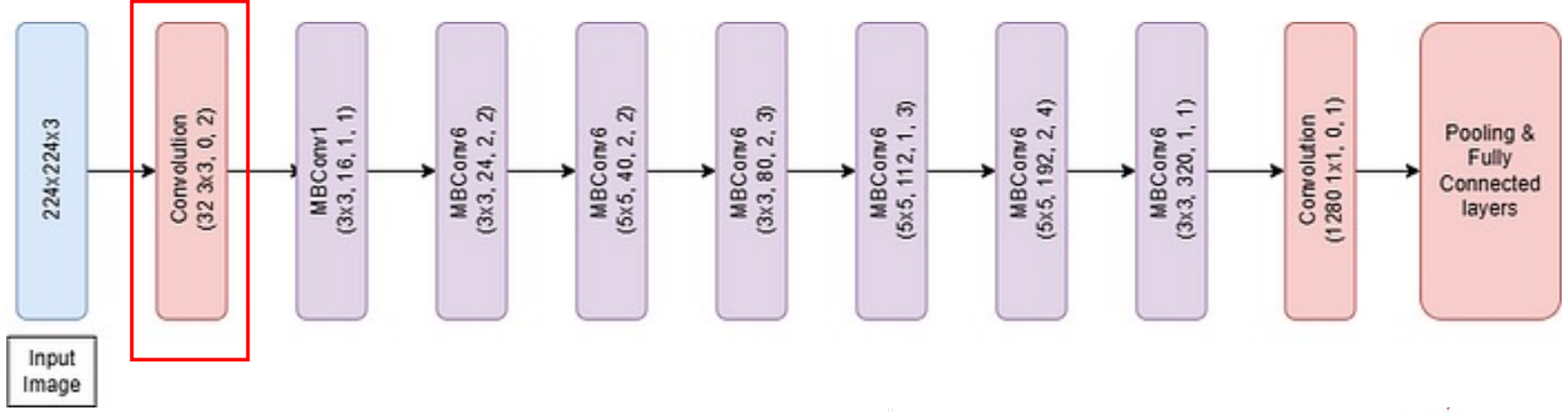
- Use running_cov and running_mean during training
- Consider maintaining running_S (=L^{-1})

Efficient Net

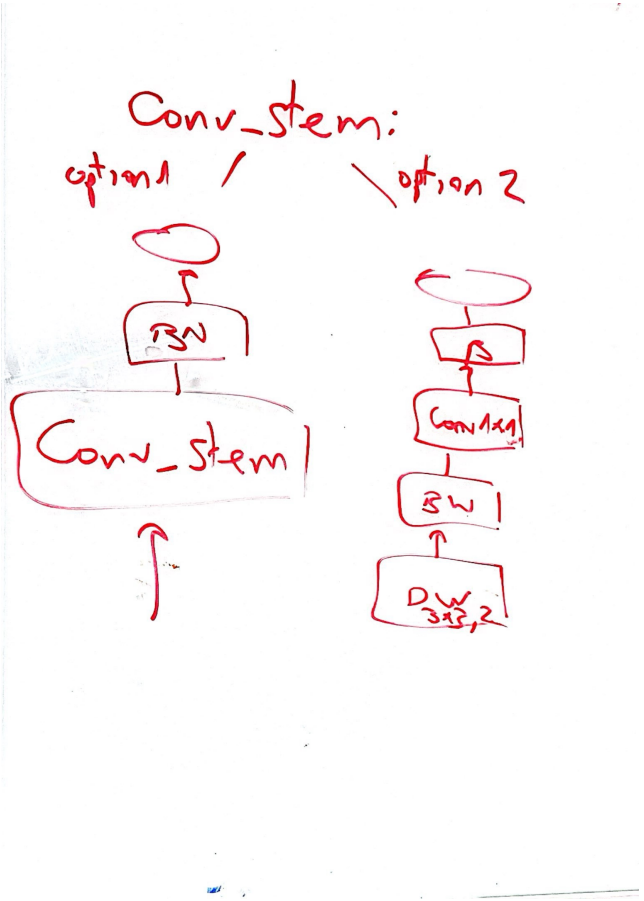


- 3 blocks/layers to handle
  - Conv_stem
  - MBConv
  - Proj_conv

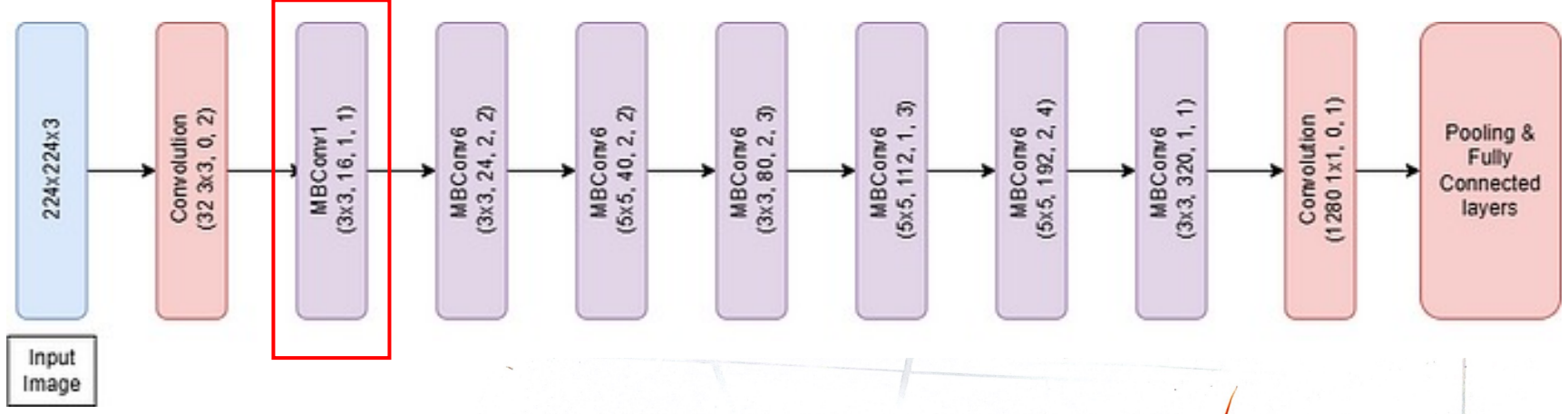
EfficientNet



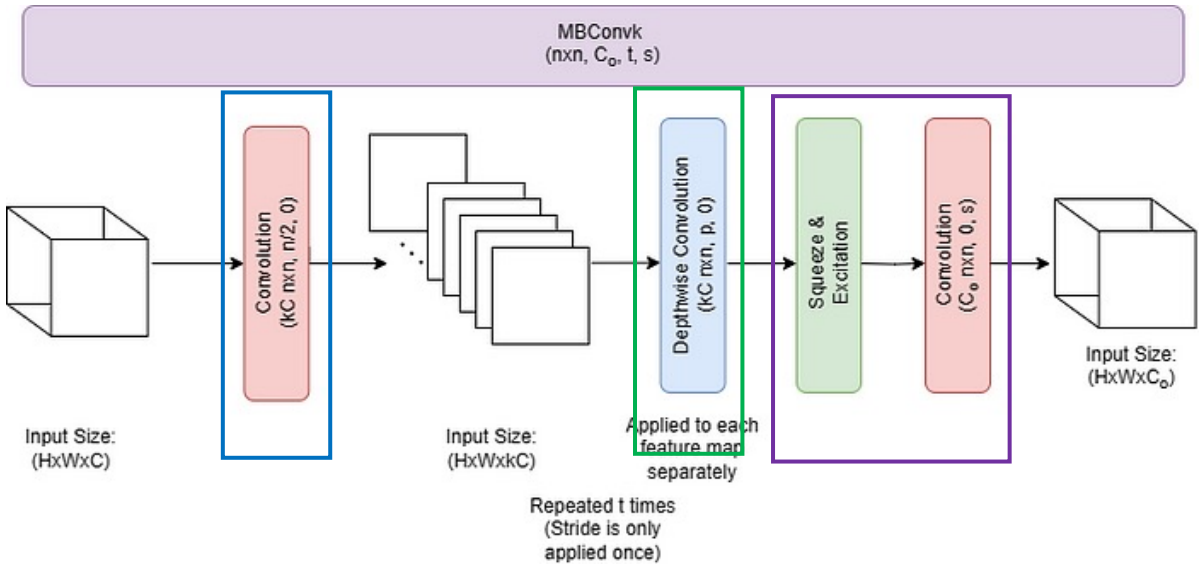
- ConvStem – 2 options
  - Type 1 (left branch) – no BW. as in original topology
  - Type 2 (right branch) – with BW

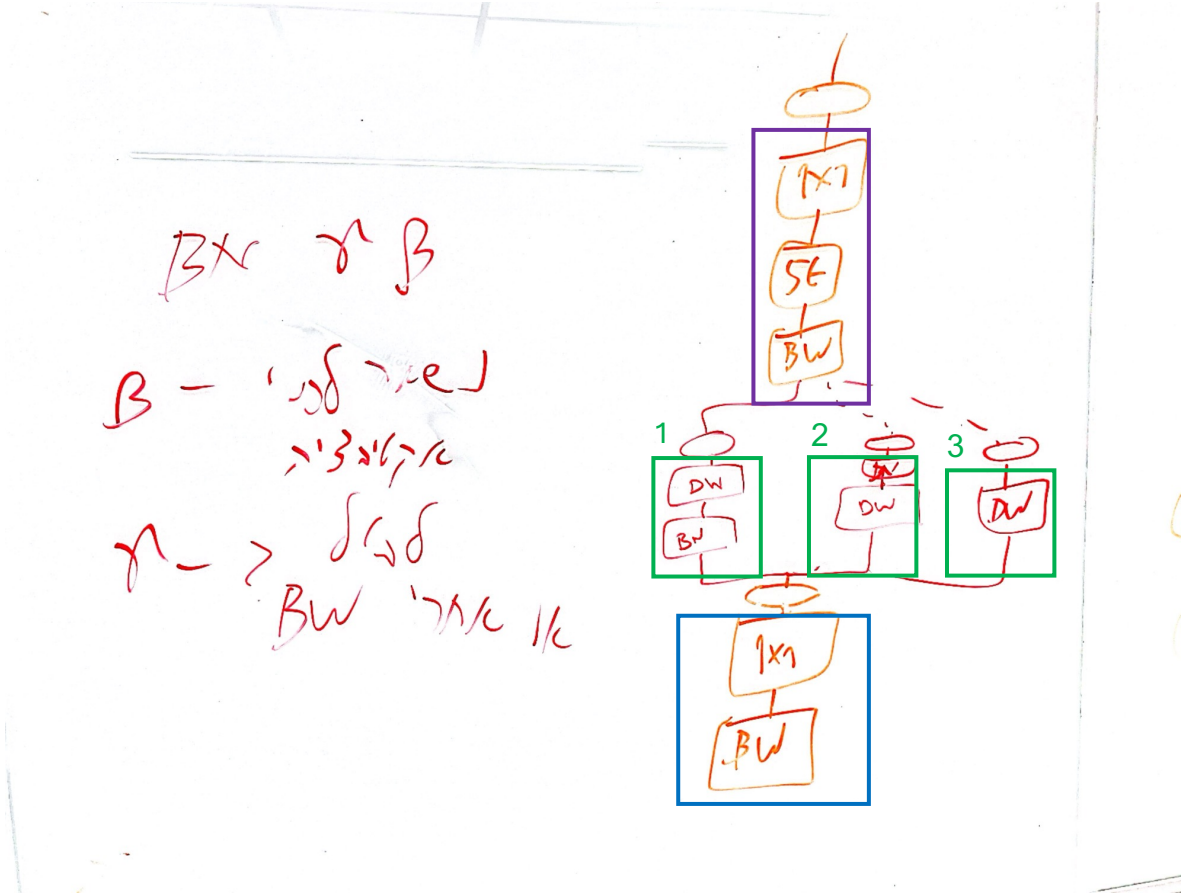


EfficientNet



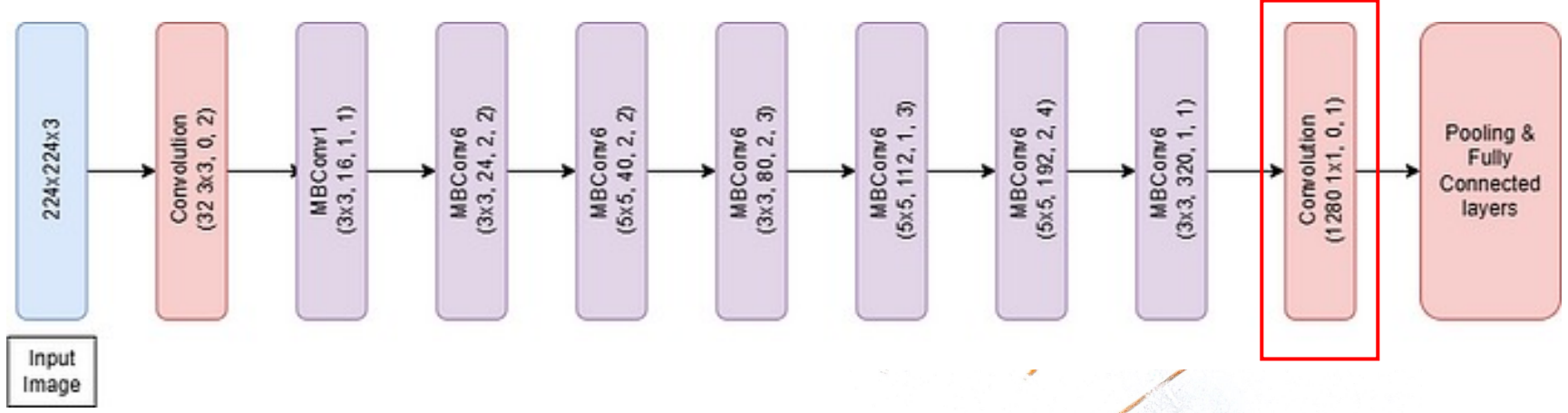
- MBConv – 3 options

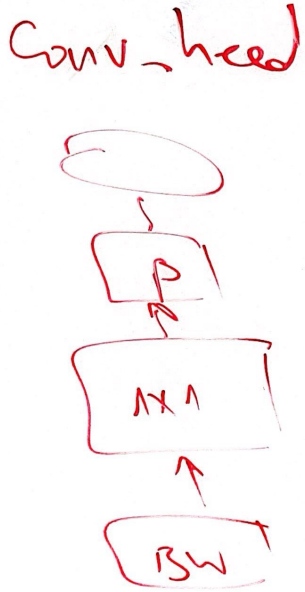




EfficientNet

- Conv head





Hyper parameter sweep

- Start on tiny imagenet
  - Mbconv block type {1,2,3}
  - Conv stem block type {1,2}
  - Define set of other potentially relevant hyperparms
  - Perform random search

- Try most successful combinations on imagenet