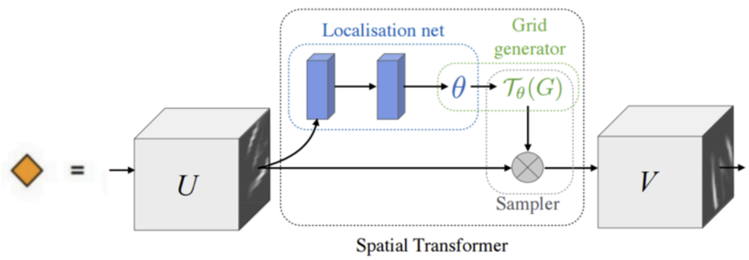
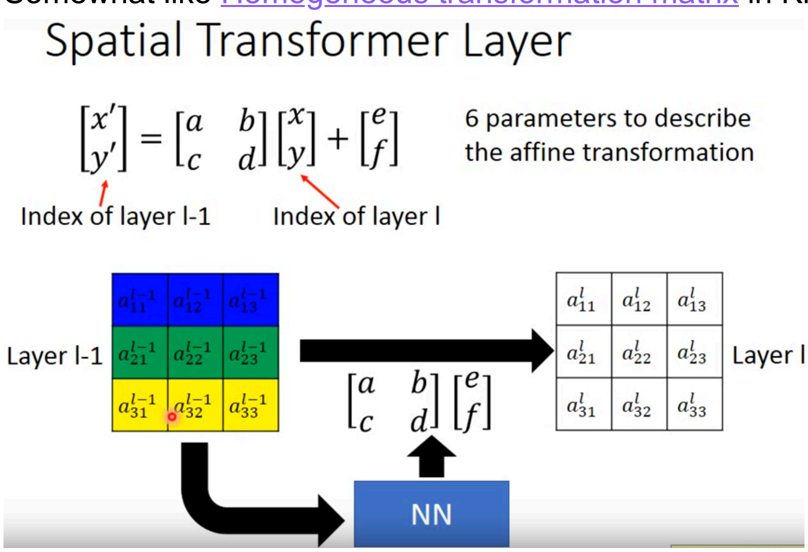
Spatial Transformer Layer

CNN is unable to address scaling and rotation,
so we use STL to transform the original picture.



1. Input Image: receive the original feature map
2. Parameter Prediction (Localisation net)
    - predict the 6 parameters for transformation
3. Coordinate Mapping (Grid generator)
    - use the predicted parameters to describe **the affine transfomation from the target to the origianl feature map**
    - Expansion, Compression, Translation, Rotation
      Somewhat like [Homogeneous transformation matrix]() in Kinamatics, but 2D version



4. Interpolation (Sampler)
   Problem 1: after Coordinate Mapping, new pixels' coordinate may not be integers, thus couldn't find an exact position
   Solution: use **Nearest-Neighbor Interpolation** to fill every pixels in the transformed picture with the RGB value of the nearset point
   Problem 2: Nearest-Neighbor Interpolation is not differentiable, thus cannot use GD

Solution: use **Bilinear Interpolation**

Interpolation

Now we can use gradient descent

$$\begin{bmatrix}1.6\\2.4\end{bmatrix} = \begin{bmatrix}0 & 0.5\\1 & 0\end{bmatrix}\begin{bmatrix}2\\2\end{bmatrix} + \begin{bmatrix}0.6\\0.4\end{bmatrix}$$

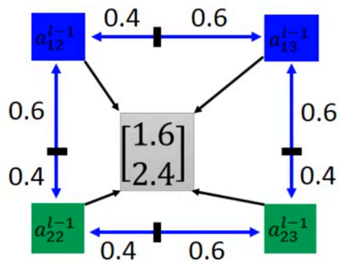
6 parameters to describe the affine transformation

ndex of layer l-1

Index of layer l

a^l_{11}	a^l_{12}	a^l_{13}
a^l_{21}	a^l_{22}	a^l_{23}
a^l_{31}	a^l_{32}	a^l_{33}

Layer l



$$\begin{aligned} a^l_{22} &= (1-0.4)\times(1-0.4)\times a^{l-1}_{22} \\ &+ (1-0.6)\times(1-0.4)\times a^{l-1}_{12} \\ &+ (1-0.6)\times(1-0.6)\times a^{l-1}_{13} \\ &+(1-\ 0.4)\times(1-0.6)\times a^{l-1}_{23} \end{aligned}$$