# Hierarchical Attribute CNNs

**Introduction**

- How can we introduce **structure** into deep networks to understand them better?
- (Mallat, 2016) proposes to perform high-dim convolutions which organise the channels
- The Dream:
  
  a) Translations along channels modulate signal attributes
  b) Layers disentangle, increasingly more complex signal attributes with depth

### An Example: Rotation covariance

![Rotation covariance](image)

\[
L_\theta x(u, \theta) = x(r_{-\theta} u, \theta - \theta)
\]

- Assume the representation encodes the rotation (rotation equivariance by \(L_\theta\))

A rotation of the input is a translation of the signal, which is a shift of \(\theta\).

**Can we generalise this beyond roto-translation?**

### Multi-dimensional Convolution

- \(u\): native spatial dimensions
- \(v_j\): signal attributes (discriminative channels)

**Operators** \(W\): convolution along spatial and channel dimensions

**Convolution** with filter \(k\) corresponds to:

\[
W_j x_{j-1} = x_{j-1} *_{u,v} k_j (u,v_{i_1},...,v_{i_M}) = \sum_{i_0} x_j (\tilde{u},\tilde{v}_{i_1},...,\tilde{v}_{i_M}) k(u - \tilde{u},v_{i_1} - \tilde{v}_{i_1},...,v_{i_M} - \tilde{v}_{i_M})
\]

**Covariant with translations along** \((v_1,...,v_j)\): \(L W x = W L x\)

### Multiscale Hierarchical CNNs

- Class of CNN where one-dimensional channel index is replaced by multi-dimensional vector of attributes: \(v = (v_1,...,v_j)\).
- Output of layer \(j\) is represented by: \(x(u,v_1,...,v_j)\)
- All linear operators \(W_j\) are convolutions over \((u,v)\), introduced above; each convolution introduces a new attribute \(v_{j+1}\) the index of the new filters, namely \(j\):

\[
x_j = \rho W_j x_{j-1}
\]

**Issues:**
- Exponential increase of parameters with depth
- Large layers, number of attributes increases
- N-dimensional convolutions are expensive

### Hierarchical Attribute CNNs

- Overcomes implementation **limitations** of Multiscale Hierarchical CNNs and introduces **increasing** invariance along attributes
- Core idea: Eliminate dependency to all attributes, but last three:

\[
x_j(v_{j-2}, v_{j-1}, v_j) \rightarrow \begin{cases} x_j(v_{j+1}, v_j, v_{j-1}) \\ \text{will be removed via an averaging in the linear operator} \end{cases}
\]

- New attributes are created, convolved along twice and eliminated

- **Good performance requires regularisation and careful implementation**

**Architecture:**

\[
x_j(u) = 2D - 3D - 5D - 5D - 5D - 5D - 5D - 5D - 5D - 5D - \sum_{(a,v)} \theta
\]

### Numerical Results

- Hierarchical Attribute CNNs dramatically reduce trainable parameters → **Organisation is effective!!**
- Improved classification performance vs. #Parameters trade-off / complexity of training, compared to FitNet or teacher student methods
- Cifar-10 classification performance on par with comparable CNNs

<table>
<thead>
<tr>
<th>Model</th>
<th>HCNN</th>
<th>HCNN(+)</th>
<th>All-CNN</th>
<th>ResNet-20</th>
<th>NiN</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Acc</td>
<td>91.23</td>
<td>92.3</td>
<td>92.75</td>
<td>91.25</td>
<td>91.20</td>
</tr>
<tr>
<td>#params</td>
<td>0.1M</td>
<td>0.3M</td>
<td>1.3M</td>
<td>0.3M</td>
<td>1.0M</td>
</tr>
</tbody>
</table>

**Investigating translations** \(L\): 

\[
\tau_{(r_{i_1})} x(u,v_{i_1}) \rightarrow \tau_{(r_{i_2})} x(u,v_{i_2}) \rightarrow ... 
\]

### Conclusions

- Highly structured: They achieve good classification performance with much less parameters than comparable CNNs
- Hierarchical Attribute CNNs give a framework to study deep network representations
- But attributes are hard to interpret!! Is there a theoretical limitation, are we missing an idea or is there something deeper?

### Code

- [GitHub](https://github.com/jhjacobsen/HierarchicalCNN)

---

**Authors**

Jörn-Henrik Jacobsen, Edouard Oyallon, Stéphane Mallat, Arnold W.M. Smeulders

**Institute of Informatics, University of Amsterdam**

**DATA, Département Informatique, Ecole Normale Supérieure/PSL Research University**