Multi-dimensional feature and data mining
Georgiou, T.

Citation

Version: Publisher's Version

License: Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden

Downloaded from: https://hdl.handle.net/1887/3214119

Note: To cite this publication please use the final published version (if applicable).
A Clifford Convolution gradients calculations

Below are all the calculations of all the equations used for the back-propagation algorithm. For all formulas, for the sake of simplicity we use the following notation:

\[ p_w : \text{weight position, } (i, j, c_{in}, c_{out}) \]
\[ p_o : \text{output position, } (i', j', c_{out}) \]
\[ p_{in} : \text{input position, } (i'', j'', c_{in}) \] (1)

During the backward pass, every arrow in Figure 5.2 should return the respective derivatives. We start the computations from step \( C_5 \) and work our way to step \( C_1 \).

\( C_5 \):
The final output \( \hat{O}_{p_o}^l \) is given by:

\[ \hat{O}_{p_o}^l = (O_{0,p_o}^l, O_{1,p_o}^l) = (O_{p_o}^l \cdot \cos (\phi_{p_o}), O_{p_o}^l \cdot \sin (\phi_{p_o})) \] (2)

Following the chain rule:

\[
\frac{\partial E}{\partial O_{p_o}^l} = \frac{\partial E}{\partial O_{0,p_o}^l} \frac{\partial O_{0,p_o}^l}{\partial O_{p_o}^l} + \frac{\partial E}{\partial O_{1,p_o}^l} \frac{\partial O_{1,p_o}^l}{\partial O_{p_o}^l} \Rightarrow
\]

\[
\frac{\partial E}{\partial O_{p_o}^l} = \delta^l_{0,C_5} \cdot \cos (\phi_{p_o}) + \delta^l_{1,C_5} \cdot \sin (\phi_{p_o})
\] (3)
\[
\frac{\partial E}{\partial \phi_{p_o}} = \frac{\partial E}{\partial O_{0,p_o}} \frac{\partial O_{0,p_o}}{\partial \phi_{p_o}} + \frac{\partial E}{\partial O_{1,p_o}} \frac{\partial O_{1,p_o}}{\partial \phi_{p_o}} \Rightarrow \\
\frac{\partial E}{\partial \phi_{p_o}} = -\delta_{0,p_o} \cdot O_{0,p_o} \cdot \sin (\phi_{p_o}) + \delta_{1,p_o} \cdot O_{1,p_o} \cdot \cos (\phi_{p_o})
\]

\[C_4:\]
\[
O_{p_o}^l = \sum_i \sum_j \sum_{c_{in}} \hat{W}_{\phi p_o p_w}^l \cdot \hat{O}_{p_{in}}^{l-1} = \sum_i \sum_j \sum_{c_{in}} \sum_{k} W_{\phi p_o p_w,k}^l \cdot O_{p_{in},k}^{l-1}
\]

where \(k\) indexes the \(x\) and \(y\) coordinates of the vectors.

\[
\frac{\partial E}{\partial \hat{W}_{\phi p_o p_w,k}^l} = \frac{\partial E}{\partial O_{p_o}^l} \frac{\partial O_{p_o}^l}{\partial \hat{W}_{\phi p_o p_w,k}^l} = \sum_{p_o \in P} \delta_{p_o}^{l,C_4} \cdot O_{p_{in},k}^{l-1}
\]

where \(\delta_{p_o}^{l,C_4}\) the prejudined angles \((\in [0, 2\pi])\) used for the specific output position.

\[C_3:\]
\[
\hat{W}_{\phi p_o}^l = rotation(\hat{W}_{p_o}^l, \phi_{p_o}^l)
\]

As mentioned above, there is a separate set of weights for every output pixel. The gradients from all sets of weights are calculated and then added to the original weight vector. For simplicity the index of \(p_o\) on the weight vectors is omitted. From equation 9 we see that there are two sets of gradients need to be computed, \(\frac{\partial E}{\partial \hat{W}_p^l}\) and \(\frac{\partial E}{\partial \phi_{p_o}}\).

Since \(\hat{W}_{\phi}^l\) are the rotated \(\hat{W}_p^l\), we set:

\[
\frac{\partial E}{\partial \hat{W}_p^l} = rotation(\frac{\partial E}{\partial \hat{W}_p^l}, -\phi_{p_o}^l)
\]
For the second set we have:

$$\frac{\partial E}{\partial \phi_{p_0}} = \sum_i \sum_j \sum_{c_{in}} \frac{\partial E}{\partial \hat{W}^l_{\phi_{p_0}} \phi_{p_0}} = \sum_i \sum_j \sum_{c_{in}} \delta_{W_{\phi_{p_0}}}^l C_3 \cdot \frac{\partial \hat{W}_l^{\phi}_{p_0}}{\partial \phi_{p_0}} \quad (11)$$

We have two options for calculating $\frac{\partial \hat{W}_l^{\phi}_{p_0}}{\partial \phi_{p_0}}$. The first is to differentiate the bilinear interpolation (at least at the points that it is differentiable), or use the precalculated rotated weights. Let $\theta$ be the quantized calculated angle $\phi$. Then:

$$\frac{\partial \hat{W}_{\theta_{p_0}}}{\partial \phi_{p_0}} = \frac{\hat{W}_{\theta+1_{p_0}} - \hat{W}_{\theta-1_{p_0}}}{2\frac{2\pi}{B}} \quad (12)$$

where $B$ is the number of predefined orientations. Have in mind that the rotation above ($\hat{W}_{\theta+1_{p_0}}$) considers only plane rotation after acquiring the in-place vector rotation $\hat{W}_\phi$.

$C_2$:

On all equations related to $C_2$, like $C_3$, the indexes $p_0$ on weight vectors are omitted.

$$\hat{W}_\phi^{l_{p_w}} = \hat{W}_{p_w}^{l} \cdot \begin{pmatrix} \cos \phi_{p_0} & \sin \phi_{p_0} \\ -\sin \phi_{p_0} & \cos \phi_{p_0} \end{pmatrix} \Leftrightarrow \begin{cases} \hat{W}_{\phi_0}^{l_{p_w}} = \hat{W}_{0,p_w}^{l} \cos \phi_{p_0} - \hat{W}_{1,p_w}^{l} \sin \phi_{p_0} \\ \hat{W}_{\phi_1}^{l_{p_w}} = \hat{W}_{0,p_w}^{l} \sin \phi_{p_0} + \hat{W}_{1,p_w}^{l} \cos \phi_{p_0} \end{cases} \quad (13)$$

As with $C_3$, there are two set of gradients to be calculated, specifically $\frac{\partial E}{\partial W_{p_w}^{l}}$ and $\frac{\partial E}{\partial \phi_{p_0}}$ where the first represents two sets, one for each direction of the vectors in $\hat{W}_l$.

$$\frac{\partial E}{\partial W_{p_w}^{l}} = \left( \frac{\partial E}{\partial W_{0,p_w}^{l}} , \frac{\partial E}{\partial W_{1,p_w}^{l}} \right) \quad (14)$$

For the two components we get:

$$\frac{\partial E}{\partial W_{0,p_w}^{l}} = \left( \frac{\partial E}{\partial W_{\phi_0}^{l_{p_w}}} \frac{\partial \hat{W}_{\phi_{0,p_w}}^{l}}{\partial \phi_{p_0}} + \frac{\partial E}{\partial \hat{W}_{\phi_1}^{l_{p_w}}} \frac{\partial \hat{W}_{\phi_{1,p_w}}^{l}}{\partial \phi_{p_0}} \right)$$

$$= \left( \frac{\partial E}{\partial W_{\phi_0}^{l_{p_w}}} \cos \phi_{p_0} + \frac{\partial E}{\partial \hat{W}_{\phi_1}^{l_{p_w}}} \sin \phi_{p_0} \right)$$

$$\frac{\partial E}{\partial W_{1,p_w}^{l}} = \left( \frac{\partial E}{\partial W_{\phi_0}^{l_{p_w}}} \frac{\partial \hat{W}_{\phi_{0,p_w}}^{l}}{\partial \phi_{p_0}} + \frac{\partial E}{\partial \hat{W}_{\phi_1}^{l_{p_w}}} \frac{\partial \hat{W}_{\phi_{1,p_w}}^{l}}{\partial \phi_{p_0}} \right)$$

$$= \left( -\frac{\partial E}{\partial W_{\phi_0}^{l_{p_w}}} \sin \phi_{p_0} + \frac{\partial E}{\partial \hat{W}_{\phi_1}^{l_{p_w}}} \cos \phi_{p_0} \right) \quad (15)$$
\[
\begin{align*}
(14), (15) \rightarrow \frac{\partial E}{\partial \vec{W}_{p_w}^l} &= \frac{\partial E}{\partial \vec{W}_{\phi_{p_w}}^l} \cdot \begin{pmatrix}
\cos (-\phi_{p_w}^l) & \sin (-\phi_{p_w}^l) \\
-\sin (-\phi_{p_w}^l) & \cos (-\phi_{p_w}^l)
\end{pmatrix} \\
&= \delta^l, C_2 \vec{\phi}_{p_w} \\
\frac{\partial E}{\partial \vec{W}_{\phi_{p_w}}^l} &= \frac{\partial E}{\partial \vec{W}_{0, p_w}^l} + \frac{\partial E}{\partial \vec{W}_{1, p_w}^l} \frac{\partial \vec{W}_{0, p_w}^l}{\partial \phi_{p_w}^l} + \frac{\partial E}{\partial \vec{W}_{1, p_w}^l} \frac{\partial \vec{W}_{1, p_w}^l}{\partial \phi_{p_w}^l}
\end{align*}
\]

\[
\frac{\partial \vec{W}_{0, p_w}^l}{\partial \phi_{p_w}^l} = -\vec{W}_{0, p_w}^l \sin \phi_{p_w}^l - \vec{W}_{1, p_w}^l \cos \phi_{p_w}^l = -\vec{W}_{1, p_w}^l
\]

\[
\frac{\partial \vec{W}_{1, p_w}^l}{\partial \phi_{p_w}^l} = \vec{W}_{0, p_w}^l \cos \phi_{p_w}^l - \vec{W}_{1, p_w}^l \sin \phi_{p_w}^l = \vec{W}_{0, p_w}^l
\]

\[
(17), (18) \rightarrow \frac{\partial E}{\partial \phi_{p_w}^l} = \sum_i \sum_j \sum_c \left( \frac{\partial E}{\partial \vec{W}_{0, p_w}^l} \frac{\partial \vec{W}_{0, p_w}^l}{\partial \phi_{p_w}^l} + \frac{\partial E}{\partial \vec{W}_{1, p_w}^l} \frac{\partial \vec{W}_{1, p_w}^l}{\partial \phi_{p_w}^l} \right)
\]

\[
= \sum_i \sum_j \sum_c (\delta^{l, C_2}_{0, p_w} \vec{W}_{0, p_w}^l + \delta^{l, C_2}_{1, p_w} \vec{W}_{1, p_w}^l)
\]

In our implementation $C_2$ and $C_3$ are considered as one operation. Moreover, we keep in memory the rotated weights $\vec{W}_\phi$ and not $\vec{W}_\phi$. Fortunately, we can approximate the gradients as of the separate operations as following:

\[
\vec{W}_\phi^l = vector\_field\_rotation(\vec{W}_\phi^l, \phi_{p_w}^l)
\]

Similarly with $C_3$:

\[
\frac{\partial E}{\partial \vec{W}_\phi^l} = vector\_field\_rotation(\frac{\partial E}{\partial \vec{W}_\phi^l}, -\phi_{p_w}^l)
\]

\[
\frac{\partial \vec{W}_{\theta + 1, p_w}^l}{\partial \phi_{p_w}^l} = \frac{\vec{W}_{\theta + 1, p_w}^l - \vec{W}_{\theta - 1, p_w}^l}{2\pi B}
\]

Unlike $C_3$, here the rotated $\vec{W}_{\theta + 1, p_w}^l$ are the complete vector field rotation with angle $\theta + 1$ from the original $\vec{W}$.
\[ C_1: \]

Let:

\[ \tan^l_{p_o} = \frac{\text{conv}^l_{0,p_o}}{\text{conv}^l_{2,p_o}} \]  

(23)

then:

\[ (5.2), (5.1) \rightarrow \phi^l_{p_o} = \arctan\left(\frac{\text{conv}^l_{2}}{\text{conv}^l_{0}}\right) = \arctan(\tan^l_{p_o}) \]  

(24)

\[ \frac{\partial E}{\partial \tan^l_{p_o}} = \frac{\partial E}{\partial \phi^l_{p_o}} \frac{1}{\partial \tan^l_{p_o}} = \frac{\partial E}{\partial \phi^l_{p_o}} \frac{1}{1 + (\tan^l_{p_o})^2} \]  

(25)

\[ \frac{\partial E}{\partial \text{conv}^l_{0}} = \frac{\partial E}{\partial \phi^l_{p_o}} \frac{1}{\partial \text{conv}^l_{0}} = \frac{\partial E}{\partial \phi^l_{p_o}} \frac{(-\text{conv}^l_{2})}{\text{conv}^l_{0}} \Rightarrow \]  

(26)

\[ \frac{\partial E}{\partial \text{conv}^l_{2}} = \frac{\partial E}{\partial \phi^l_{p_o}} \frac{1}{\partial \text{conv}^l_{2}} = \frac{\partial E}{\partial \phi^l_{p_o}} \frac{1}{\text{conv}^l_{0}} \Rightarrow \]  

(27)

From Equation 5.2 we see that \( \text{conv}^l_{0} \) is the conventional convolutional operation, meaning that the derivatives are the standard derivatives used in all CNN works. For \( \text{conv}^l_{2} \) we have:

\[ \frac{\partial E}{\partial W^l_{0,p_w}} = \sum_{i'} \sum_{j'} \frac{\partial E}{\partial \text{conv}^l_{2}} \frac{\partial \text{conv}^l_{2}}{\partial W^l_{0,p_w}} = \sum_{i'} \sum_{j'} \frac{\partial E}{\partial \text{conv}^l_{2}} O^l_{1,p_{in}} \]  

(28)

\[ \frac{\partial E}{\partial W^l_{1,p_w}} = \sum_{i'} \sum_{j'} \frac{\partial E}{\partial \text{conv}^l_{2}} \frac{\partial \text{conv}^l_{2}}{\partial W^l_{1,p_w}} = \sum_{i'} \sum_{j'} \frac{\partial E}{\partial \text{conv}^l_{2}} (-O^l_{0,p_{in}}) \]

Similarly:

\[ \frac{\partial E}{\partial O^l_{0,p_{in}}} = \sum_{i} \sum_{j} \frac{\partial E}{\partial \text{conv}^l_{2}} \frac{\partial \text{conv}^l_{2}}{\partial O^l_{0,p_{in}}} = \sum_{i} \sum_{j} \frac{\partial E}{\partial \text{conv}^l_{2}} (-O^l_{1,p_{in}}) \]  

(29)

\[ \frac{\partial E}{\partial O^l_{1,p_{in}}} = \sum_{i} \sum_{j} \frac{\partial E}{\partial \text{conv}^l_{2}} \frac{\partial \text{conv}^l_{2}}{\partial O^l_{1,p_{in}}} = \sum_{i} \sum_{j} \frac{\partial E}{\partial \text{conv}^l_{2}} W^l_{0,p_w} \]

For each output pixel a separate weight vector was calculated and thus different gradients as well, i.e., \( \left( \frac{\partial E}{\partial \tilde{W}^l_{p_o}} \right) \). The final result is given by adding the \( \left( \frac{\partial E}{\partial \tilde{W}^l_{p_o}} \right) \) for all \( p_o \).
B Table of abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>two dimensions/dimensional</td>
</tr>
<tr>
<td>3D</td>
<td>three dimensions/dimensional</td>
</tr>
<tr>
<td>3DBRIEF</td>
<td>3D BRIEF</td>
</tr>
<tr>
<td>3DLBP</td>
<td>3D LBP</td>
</tr>
<tr>
<td>3DORB</td>
<td>3D ORB</td>
</tr>
<tr>
<td>3DSC</td>
<td>3D SC</td>
</tr>
<tr>
<td>4D</td>
<td>four dimensions/dimensional</td>
</tr>
<tr>
<td>Adam</td>
<td>adaptive moment estimation</td>
</tr>
<tr>
<td>AE</td>
<td>auto-encoder</td>
</tr>
<tr>
<td>AGAST</td>
<td>adaptive and generic accelerated segment test</td>
</tr>
<tr>
<td>AlexNet</td>
<td>Alex Network</td>
</tr>
<tr>
<td>AMT</td>
<td>Amazon mechanical turk</td>
</tr>
<tr>
<td>ANN</td>
<td>artificial neural network</td>
</tr>
<tr>
<td>APC</td>
<td>Amazon picking challenge</td>
</tr>
<tr>
<td>API</td>
<td>application programming interface</td>
</tr>
<tr>
<td>avacc</td>
<td>meanIU</td>
</tr>
<tr>
<td>B3DO</td>
<td>Berkley 3D Objects</td>
</tr>
<tr>
<td>BN</td>
<td>batch normalization</td>
</tr>
<tr>
<td>BoF</td>
<td>bag of features</td>
</tr>
<tr>
<td>BoW</td>
<td>bag of words</td>
</tr>
<tr>
<td>BPTT</td>
<td>back propagation through time</td>
</tr>
<tr>
<td>BRAND</td>
<td>binary robust appearance and normal descriptor</td>
</tr>
<tr>
<td>BRIEF</td>
<td>binary robust independent elementary features</td>
</tr>
<tr>
<td>BRISK</td>
<td>binary robust invariant scale keypoint</td>
</tr>
<tr>
<td>BRoPH</td>
<td>binary rotational projection histogram</td>
</tr>
<tr>
<td>C3D</td>
<td>convolutional 3D</td>
</tr>
<tr>
<td>CAD</td>
<td>computer-aided design</td>
</tr>
<tr>
<td>CAE</td>
<td>convolutional AE</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Explanation</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CBCT</td>
<td>cone beam computed tomography</td>
</tr>
<tr>
<td>cc</td>
<td>Clifford convolution</td>
</tr>
<tr>
<td>CFD</td>
<td>computational fluid dynamics</td>
</tr>
<tr>
<td>CFN</td>
<td>convolutional fusion network</td>
</tr>
<tr>
<td>Charades-STA</td>
<td>Charades sentence temporal annotations</td>
</tr>
<tr>
<td>CHMM</td>
<td>coupled HMM</td>
</tr>
<tr>
<td>CIFAR</td>
<td>Canadian institute for advanced research</td>
</tr>
<tr>
<td>CL</td>
<td>convolutional layer</td>
</tr>
<tr>
<td>clacc</td>
<td>classification accuracy</td>
</tr>
<tr>
<td>CNN</td>
<td>convolutional neural network</td>
</tr>
<tr>
<td>COCO</td>
<td>common objects in context</td>
</tr>
<tr>
<td>convGRBM</td>
<td>convolutional GRBM</td>
</tr>
<tr>
<td>CPU</td>
<td>central processing unit</td>
</tr>
<tr>
<td>CRF</td>
<td>conditional random field</td>
</tr>
<tr>
<td>CT</td>
<td>computerized tomography</td>
</tr>
<tr>
<td>DAE</td>
<td>denoising AE</td>
</tr>
<tr>
<td>DB</td>
<td>database</td>
</tr>
<tr>
<td>DBM</td>
<td>deep Boltzmann machines</td>
</tr>
<tr>
<td>DBN</td>
<td>deep belief network</td>
</tr>
<tr>
<td>D-CNN</td>
<td>deep CNN</td>
</tr>
<tr>
<td>DE</td>
<td>dense sampling</td>
</tr>
<tr>
<td>DEM</td>
<td>deep energy model</td>
</tr>
<tr>
<td>DenseNet</td>
<td>dense network</td>
</tr>
<tr>
<td>DiDeMo</td>
<td>distinct describable moments</td>
</tr>
<tr>
<td>DL</td>
<td>deep learning</td>
</tr>
<tr>
<td>DNN</td>
<td>deep neural network</td>
</tr>
<tr>
<td>DoG</td>
<td>difference of Gaussians</td>
</tr>
<tr>
<td>DoF</td>
<td>degrees of freedom</td>
</tr>
<tr>
<td>DS</td>
<td>direction specific</td>
</tr>
<tr>
<td>DSN</td>
<td>deeply supervised nets</td>
</tr>
<tr>
<td>DSTIP</td>
<td>depth STIP</td>
</tr>
<tr>
<td>ED</td>
<td>elevation descriptor</td>
</tr>
<tr>
<td>ELU</td>
<td>exponential linear unit</td>
</tr>
<tr>
<td>EMK</td>
<td>efficient match kernel</td>
</tr>
<tr>
<td>EVD</td>
<td>eigenvalue decomposition</td>
</tr>
<tr>
<td>FAST</td>
<td>features from accelerated segment test</td>
</tr>
<tr>
<td>FC</td>
<td>fully connected</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Explanation</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>FCN</td>
<td>fully convolutional networks</td>
</tr>
<tr>
<td>FCVID</td>
<td>Fudan-Columbia video dataset</td>
</tr>
<tr>
<td>FMS</td>
<td>full modality specific</td>
</tr>
<tr>
<td>FPFH</td>
<td>fast PFH</td>
</tr>
<tr>
<td>FREAK</td>
<td>fast retina keypoint</td>
</tr>
<tr>
<td>fus-CNN</td>
<td>fusion CNN</td>
</tr>
<tr>
<td>fwavacc</td>
<td>frequency weighted average accuracy</td>
</tr>
<tr>
<td>GAH</td>
<td>geometric attribute histograms</td>
</tr>
<tr>
<td>GAN</td>
<td>generative adversarial network</td>
</tr>
<tr>
<td>GFU</td>
<td>gated fusion unit</td>
</tr>
<tr>
<td>GNN</td>
<td>graph neural network</td>
</tr>
<tr>
<td>GPU</td>
<td>graphics processing unit</td>
</tr>
<tr>
<td>GRBM</td>
<td>gated RBM</td>
</tr>
<tr>
<td>GRU</td>
<td>gated recurrent unit</td>
</tr>
<tr>
<td>GT</td>
<td>ground truth</td>
</tr>
<tr>
<td>HAR</td>
<td>human action recognition</td>
</tr>
<tr>
<td>Harris3D</td>
<td>Harris 3D</td>
</tr>
<tr>
<td>HBN</td>
<td>half layers batch normalized</td>
</tr>
<tr>
<td>HCRF</td>
<td>hidden CRF</td>
</tr>
<tr>
<td>HHA</td>
<td>horizontal disparity, height above ground, angle the pixel's local surface normal makes with the inferred gravity direction</td>
</tr>
<tr>
<td>HKDE</td>
<td>hierarchical KDE</td>
</tr>
<tr>
<td>HKS</td>
<td>heat kernel signature</td>
</tr>
<tr>
<td>HMC</td>
<td>hidden Markov chain</td>
</tr>
<tr>
<td>HMDB51</td>
<td>human motion database</td>
</tr>
<tr>
<td>HMM</td>
<td>hidden Markov model</td>
</tr>
<tr>
<td>HMP</td>
<td>hierarchical matching pursuit</td>
</tr>
<tr>
<td>HOF</td>
<td>histogram of flow</td>
</tr>
<tr>
<td>HOG</td>
<td>histogram of oriented gradients</td>
</tr>
<tr>
<td>HON</td>
<td>histogram of surface normals</td>
</tr>
<tr>
<td>HON4D</td>
<td>HON 4D</td>
</tr>
<tr>
<td>HOPC</td>
<td>histogram of principal components</td>
</tr>
<tr>
<td>HSMM</td>
<td>hidden semi-Markov model</td>
</tr>
<tr>
<td>I3D</td>
<td>inflated 3D CNN</td>
</tr>
<tr>
<td>IDT</td>
<td>improved dense trajectories</td>
</tr>
<tr>
<td>IP</td>
<td>interest point</td>
</tr>
<tr>
<td>KDE</td>
<td>kernel descriptor</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Explanation</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>kd-tree</td>
<td>k dimensional tree</td>
</tr>
<tr>
<td>KITTI</td>
<td>? (not mentioned in the work that proposes it [98])</td>
</tr>
<tr>
<td>KLT</td>
<td>Kanade Lucas-Tomasi</td>
</tr>
<tr>
<td>k-NN</td>
<td>k nearest neighbors</td>
</tr>
<tr>
<td>kSVM</td>
<td>kernel SVM</td>
</tr>
<tr>
<td>KTH</td>
<td>Royal institute of technology, Stockholm</td>
</tr>
<tr>
<td>LBP</td>
<td>local binary pattern</td>
</tr>
<tr>
<td>LeNet</td>
<td>LeCun network</td>
</tr>
<tr>
<td>LFD</td>
<td>light field descriptor</td>
</tr>
<tr>
<td>LFSH</td>
<td>local feature statistics histogram</td>
</tr>
<tr>
<td>LiDAR</td>
<td>light detection and ranging</td>
</tr>
<tr>
<td>LINE</td>
<td>linearizing the memory</td>
</tr>
<tr>
<td>LINEMOD</td>
<td>multimodal LINE</td>
</tr>
<tr>
<td>linSVM</td>
<td>linear SVM</td>
</tr>
<tr>
<td>LN</td>
<td>locally connected</td>
</tr>
<tr>
<td>LRCN</td>
<td>long-term recurrent CNN</td>
</tr>
<tr>
<td>LReLU</td>
<td>leaky ReLU</td>
</tr>
<tr>
<td>LRF</td>
<td>local reference frame</td>
</tr>
<tr>
<td>LSP</td>
<td>local surface patch</td>
</tr>
<tr>
<td>LSTM</td>
<td>long short-term memory node</td>
</tr>
<tr>
<td>LSTM-CF</td>
<td>LSTM context fusion</td>
</tr>
<tr>
<td>LTC</td>
<td>long temporal convolutional network</td>
</tr>
<tr>
<td>LTP</td>
<td>local trinary pattern</td>
</tr>
<tr>
<td>MAE</td>
<td>mean absolute error</td>
</tr>
<tr>
<td>MD</td>
<td>multiple dictionary</td>
</tr>
<tr>
<td>meanIU</td>
<td>mean intersection over union</td>
</tr>
<tr>
<td>MK-MMD</td>
<td>multiple kernel maximum mean discrepancy</td>
</tr>
<tr>
<td>MLP</td>
<td>multi-layer perceptron</td>
</tr>
<tr>
<td>MMF</td>
<td>multi modal feature fusion</td>
</tr>
<tr>
<td>MNIST</td>
<td>modified national institute of standards and technology</td>
</tr>
<tr>
<td>MO-AniProbing</td>
<td>multi orientation anisotropic probing</td>
</tr>
<tr>
<td>mp</td>
<td>max pooling</td>
</tr>
<tr>
<td>MR</td>
<td>magnetic resonance</td>
</tr>
<tr>
<td>MRF</td>
<td>Markov random field</td>
</tr>
<tr>
<td>MRI</td>
<td>magnetic resonance imaging</td>
</tr>
<tr>
<td>MVCNN</td>
<td>multi view CNN</td>
</tr>
<tr>
<td>MVD</td>
<td>multi-view depth</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Explanation</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>NaN</td>
<td>not a number</td>
</tr>
<tr>
<td>NBN</td>
<td>no BN</td>
</tr>
<tr>
<td>NiN</td>
<td>network in network</td>
</tr>
<tr>
<td>nl</td>
<td>norm loss</td>
</tr>
<tr>
<td>NN</td>
<td>nearest neighbor</td>
</tr>
<tr>
<td>NNDR</td>
<td>nearest neighbor distance ratio</td>
</tr>
<tr>
<td>NYU</td>
<td>New York University</td>
</tr>
<tr>
<td>NYUv2</td>
<td>NYU version 2</td>
</tr>
<tr>
<td>OGH</td>
<td>oriented gradient histograms</td>
</tr>
<tr>
<td>OLM</td>
<td>orthogonal linear module</td>
</tr>
<tr>
<td>ONI</td>
<td>orthogonalization using Newton’s iteration</td>
</tr>
<tr>
<td>op</td>
<td>orientation pooling</td>
</tr>
<tr>
<td>ORB</td>
<td>oriented FAST and rotated BRIEF</td>
</tr>
<tr>
<td>ORION</td>
<td>orientation boosted voxel net</td>
</tr>
<tr>
<td>ORN</td>
<td>orientation response network</td>
</tr>
<tr>
<td>PA-LSTM</td>
<td>part-aware LSTM</td>
</tr>
<tr>
<td>PBWN</td>
<td>projection based weight normalization</td>
</tr>
<tr>
<td>PCA</td>
<td>principal component analysis</td>
</tr>
<tr>
<td>PELU</td>
<td>parametric ELU</td>
</tr>
<tr>
<td>PFH</td>
<td>point feature histogram</td>
</tr>
<tr>
<td>pixacc</td>
<td>pixel accuracy</td>
</tr>
<tr>
<td>PPF</td>
<td>point pair feature</td>
</tr>
<tr>
<td>PReLU</td>
<td>parametric ReLU</td>
</tr>
<tr>
<td>PSB</td>
<td>Princeton shape benchmark</td>
</tr>
<tr>
<td>PSG</td>
<td>polygonal surface geometry</td>
</tr>
<tr>
<td>RA</td>
<td>reference angle</td>
</tr>
<tr>
<td>RANSAC</td>
<td>random sample consensus</td>
</tr>
<tr>
<td>RAS</td>
<td>Reynolds-averaged simulation</td>
</tr>
<tr>
<td>RBM</td>
<td>restricted Boltzmann machine</td>
</tr>
<tr>
<td>R-CNN</td>
<td>regions with CNN features</td>
</tr>
<tr>
<td>RDF</td>
<td>randomized decision forest</td>
</tr>
<tr>
<td>RDF-Net</td>
<td>RGB-D fusion network</td>
</tr>
<tr>
<td>ReLU</td>
<td>rectified linear unit</td>
</tr>
<tr>
<td>ResBlock</td>
<td>residual block</td>
</tr>
<tr>
<td>ResNet</td>
<td>residual network</td>
</tr>
<tr>
<td>RF</td>
<td>random forest</td>
</tr>
<tr>
<td>RFB</td>
<td>residual fusion block</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Explanation</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>RGB</td>
<td>Red-Green-Blue</td>
</tr>
<tr>
<td>RGB-D</td>
<td>Red-Green-Blue-Depth</td>
</tr>
<tr>
<td>RI-LBC</td>
<td>rotation invariant local binary convolution</td>
</tr>
<tr>
<td>RMSE</td>
<td>root mean square error</td>
</tr>
<tr>
<td>RNN</td>
<td>recurrent neural network</td>
</tr>
<tr>
<td>Rohr3D</td>
<td>Karl Rohr 3D</td>
</tr>
<tr>
<td>RoSP</td>
<td>rotational projection statistics</td>
</tr>
<tr>
<td>RotEqNet</td>
<td>rotation equivariant vector field network</td>
</tr>
<tr>
<td>RQ</td>
<td>research question</td>
</tr>
<tr>
<td>RSM</td>
<td>rotational silhouette map</td>
</tr>
<tr>
<td>SC</td>
<td>shape context</td>
</tr>
<tr>
<td>SD</td>
<td>single dictionary</td>
</tr>
<tr>
<td>SDH</td>
<td>spatial distribution histograms</td>
</tr>
<tr>
<td>SF</td>
<td>sparse fusion</td>
</tr>
<tr>
<td>SFCNN</td>
<td>steerable filter CNN</td>
</tr>
<tr>
<td>SfM</td>
<td>structure from motion</td>
</tr>
<tr>
<td>SGD</td>
<td>stochastic gradient descent</td>
</tr>
<tr>
<td>SHOT</td>
<td>signature of histograms of orientation</td>
</tr>
<tr>
<td>SHREC</td>
<td>shape retrieval contest</td>
</tr>
<tr>
<td>SI</td>
<td>spin image</td>
</tr>
<tr>
<td>SIFT</td>
<td>scale invariant feature transform</td>
</tr>
<tr>
<td>SI-HKS</td>
<td>SI HKS</td>
</tr>
<tr>
<td>SISI</td>
<td>scale invariant SI</td>
</tr>
<tr>
<td>SLAM</td>
<td>simultaneous localization and mapping</td>
</tr>
<tr>
<td>SP</td>
<td>superpixel</td>
</tr>
<tr>
<td>SPN</td>
<td>scalar field processing network</td>
</tr>
<tr>
<td>SRIP</td>
<td>spectral restricted isometry property</td>
</tr>
<tr>
<td>SSCD</td>
<td>spatial structure circular descriptor</td>
</tr>
<tr>
<td>SSD</td>
<td>sum of squared differences</td>
</tr>
<tr>
<td>SSMA</td>
<td>self-supervised model adaptation</td>
</tr>
<tr>
<td>SSVM</td>
<td>structural SVM</td>
</tr>
<tr>
<td>std</td>
<td>standard deviation</td>
</tr>
<tr>
<td>STIP</td>
<td>spatio-temporal interest point</td>
</tr>
<tr>
<td>ST-LSTM</td>
<td>spatio-temporal LSTM</td>
</tr>
<tr>
<td>STN</td>
<td>spatial transform networks</td>
</tr>
<tr>
<td>SUN</td>
<td>scene understanding</td>
</tr>
<tr>
<td>SUN-CG</td>
<td>? (not mentioned in the work that proposes it [346])</td>
</tr>
</tbody>
</table>
## Acknowledgements

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SURF</td>
<td>speeded up robust features</td>
</tr>
<tr>
<td>SVD</td>
<td>singular value decomposition</td>
</tr>
<tr>
<td>SVM</td>
<td>support vector machine</td>
</tr>
<tr>
<td>SYNTHIA</td>
<td>synthetic collection of imagery and annotations</td>
</tr>
<tr>
<td>TACoS</td>
<td>textually annotated cooking scenes</td>
</tr>
<tr>
<td>TDD</td>
<td>trajectory pooled deep convolutional descriptors</td>
</tr>
<tr>
<td>THRIFT</td>
<td>? (not mentioned in the work that proposes it [90])</td>
</tr>
<tr>
<td>TI</td>
<td>transformation invariant</td>
</tr>
<tr>
<td>TOLDI</td>
<td>triple orthogonal local depth images</td>
</tr>
<tr>
<td>Tri-SI</td>
<td>Tri-Spin-Image</td>
</tr>
<tr>
<td>UCF</td>
<td>university of central Florida</td>
</tr>
<tr>
<td>UMAM</td>
<td>unified model of appearance and motion</td>
</tr>
<tr>
<td>US</td>
<td>ultrasound</td>
</tr>
<tr>
<td>VC</td>
<td>velocity coherent</td>
</tr>
<tr>
<td>V-FAST</td>
<td>video FAST</td>
</tr>
<tr>
<td>VFT</td>
<td>vector field topology</td>
</tr>
<tr>
<td>VGG</td>
<td>? (not mentioned in the work that proposes it [338])</td>
</tr>
<tr>
<td>VPN</td>
<td>vector processing network</td>
</tr>
<tr>
<td>VRN</td>
<td>Voxelception ResNet</td>
</tr>
<tr>
<td>wd</td>
<td>weight decay</td>
</tr>
<tr>
<td>WKS</td>
<td>wave kernel signature</td>
</tr>
<tr>
<td>WN</td>
<td>weight normalization</td>
</tr>
<tr>
<td>WRN</td>
<td>wide ResNet</td>
</tr>
<tr>
<td>YCB</td>
<td>Yale-CMU-Berkeley</td>
</tr>
<tr>
<td>YFCC100M</td>
<td>Yahoo Flickr creative commons 100 million</td>
</tr>
</tbody>
</table>


[59] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In Proceedings of
the IEEE conference on computer vision and pattern recognition (CVPR), pages 113–123, 2019.


network training by reducing internal covariate shift. In Proceedings of the
32nd International Conference on Machine Learning (ICML), volume 37, pages

network training by reducing internal covariate shift. In Proceedings of the
Omnipress, 2015.

[155] L. Helman J. and Hesselink L. Representation and display of vector field topo-

[156] L. Helman J. and Hesselink L. Visualizing vector field topology in fluid flows.

2025, 2015.

[158] Allison Janoch, Sergey Karayev, Yangqing Jia, Jonathan T Barron, Mario Fritz,
Kate Saenko, and Trevor Darrell. A category-level 3d object dataset: Putting
the kinect to work. In Andrea Fossati, Juergen Gall, Helmut Grabner, Xiaofeng
Ren, and Kurt Konolige, editors, Consumer Depth Cameras for Computer Vision,

[159] Kevin Jarrett, Koray Kavukcuoglu, Yann LeCun, et al. What is the best multi-
stage architecture for object recognition? In IEEE 12th international conference

[160] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural net-
works for human action recognition. IEEE transactions on pattern analysis and

Exploiting feature and class relationships in video categorization with regular-
ized deep neural networks. IEEE transactions on pattern analysis and machine

[162] Yun Jiang, Stephen Moseson, and Ashutosh Saxena. Efficient grasping from
rgbd images: Learning using a new rectangle representation. In IEEE interna-
tional conference on robotics and automation (ICRA), pages 3304–3311. IEEE,
2011.


[359] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich,


