1. Introduction

Spatial Transformer layers\(^{[1]}\) (ST-layer) allow neural networks to be **invariant** to large spatial transformation by learning input-dependent transformations.

**Problem:** Current implementations support transformations that are either too restrictive e.g. affine or homographic maps, and/or destructive maps, such as thin plate splines (TPS). These kind of transformation often leads to unstable optimization of networks.

2. Method - CPAB transformations

**Our solution:** Incorporate differentiable flexible diffeomorphisms (namely, invertible differentiable transformations with a differentiable inverse) in ST-networks

CPAB transformations\(^{[2-3]}\)

1. Divide image domain into **Piecewise** tessellation
2. Equip each cell with an **Affine** velocity field
3. Enforce constraints to make **Continuous** velocity field \(v^\theta\)
4. **Based** on \(v^\theta\), solve an integral equation (numerically)

\[
\phi^\theta(x; 1) = x + \int_0^x v^\theta(\phi^\theta(x; r)) \, dr
\]

We have implemented an efficient algorithm for solving the integral in Tensorflow+CUDA.

Since the velocity field \(v^\theta\) is Lipschitz continuous, our mapping \(\phi^\theta(x)\) is a diffeomorphism.

3. Experiments

Selection of evaluated models:
- Standard CNN
- CNN with an ST-Affine layer
- CNN with an ST-TPS layer
- CNN with an ST-CPAB layer
- CNN with an ST-Affine+CPAB layer

<table>
<thead>
<tr>
<th></th>
<th>Restricted LFW</th>
<th>CelebA</th>
<th>Unrestricted LFW</th>
</tr>
</thead>
<tbody>
<tr>
<td>No transformer</td>
<td>0.788</td>
<td>0.712</td>
<td>0.893</td>
</tr>
<tr>
<td>ST-Affine</td>
<td>0.840</td>
<td>0.734</td>
<td>0.912</td>
</tr>
<tr>
<td>ST-TPS</td>
<td>0.851</td>
<td>0.742</td>
<td>0.921</td>
</tr>
<tr>
<td>ST-CPAB</td>
<td>0.878</td>
<td>0.751</td>
<td>0.936</td>
</tr>
<tr>
<td>ST-Affine+CPAB</td>
<td>0.893</td>
<td>0.756</td>
<td>0.954</td>
</tr>
<tr>
<td>State-of-the-art</td>
<td><strong>0.958</strong></td>
<td><strong>0.870</strong></td>
<td><strong>0.955</strong></td>
</tr>
</tbody>
</table>

Learning facial transformations mainly zoom in on the object in focus. The two leading principal components contribute to a vertical and horizontal translation of the images.

4. Conclusion

The proposed diffeomorphic transformation have shown:
- Good optimization properties compared to standard ST transformations
- Intuitive “squarification” transformation, given insight into possible simple data augmentation scheme
- Performance gain facial benchmark tasks
- State-of-the-art on unrestricted LFW dataset\(^{[4]}\)

References


