Computer Vision: Summary and Discussion

Computer Vision
CS 543 / ECE 549
University of Illinois

Derek Hoiem
HW 5 – PCA/FLD

• Why did training with subsets 1+5 (vs. subset 1 only) make PCA worse but FLD better?

<table>
<thead>
<tr>
<th>Method (train set)</th>
<th>Subset 1</th>
<th>Subset 2</th>
<th>Subset 3</th>
<th>Subset 4</th>
<th>Subset 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA (S1) (d=9/30)</td>
<td>0/0</td>
<td>0/0</td>
<td>0.225/0.042</td>
<td>0.664/0.564</td>
<td>0.858/0.774</td>
</tr>
<tr>
<td>FLD (S1) (c=10/31)</td>
<td>0/0</td>
<td>0/0</td>
<td>0.025/0.025</td>
<td>0.457/0.457</td>
<td>0.874/0.874</td>
</tr>
<tr>
<td>PCA (S1+S5) (d=9/30)</td>
<td>0/0</td>
<td>0.167/0</td>
<td>0.725/0.342</td>
<td>0.693/0.289</td>
<td>0/0</td>
</tr>
<tr>
<td>FLD (S1+S5) (c=10/31)</td>
<td>0/0</td>
<td>0/0</td>
<td>0/0</td>
<td>0.014/0.028</td>
<td>0/0</td>
</tr>
</tbody>
</table>
HW 5 – Deep Nets

Edward Xue (0.654)
- Network-in-network
- Careful learning rate tuning

Anish Shenoy (0.638)
- Batch normalization
- Xavier initialization
- Learning rate tuning

<table>
<thead>
<tr>
<th>Layer</th>
<th>type</th>
<th>Input</th>
<th>Filter</th>
<th>stride</th>
<th>pad</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>conv</td>
<td>32x32x3x200</td>
<td>5x5x3x384</td>
<td>1</td>
<td>2</td>
<td>32x32x384x200</td>
</tr>
<tr>
<td>2</td>
<td>relu</td>
<td>32x32x384x200</td>
<td>Max(0,x)</td>
<td>1</td>
<td>0</td>
<td>32x32x384x200</td>
</tr>
<tr>
<td>3</td>
<td>conv</td>
<td>32x32x384x200</td>
<td>1x1x384x320</td>
<td>1</td>
<td>0</td>
<td>32x32x320x200</td>
</tr>
<tr>
<td>4</td>
<td>relu</td>
<td>32x32x320x200</td>
<td>Max(0,x)</td>
<td>1</td>
<td>0</td>
<td>32x32x320x200</td>
</tr>
<tr>
<td>5</td>
<td>conv</td>
<td>32x32x320x200</td>
<td>1x1x320x192</td>
<td>1</td>
<td>0</td>
<td>32x32x192x200</td>
</tr>
<tr>
<td>6</td>
<td>relu</td>
<td>32x32x192x200</td>
<td>Max(0,x)</td>
<td>1</td>
<td>0</td>
<td>32x32x192x200</td>
</tr>
<tr>
<td>7</td>
<td>pool (max)</td>
<td>32x32x192x200</td>
<td>3x3</td>
<td>2</td>
<td>0</td>
<td>15x15x192x200</td>
</tr>
<tr>
<td>8</td>
<td>dropout (0.5)</td>
<td>15x15x192x200</td>
<td>5x5x3</td>
<td>2</td>
<td>0</td>
<td>7x7x192x200</td>
</tr>
<tr>
<td>9</td>
<td>conv</td>
<td>15x15x192x200</td>
<td>5x5x192x384</td>
<td>1</td>
<td>2</td>
<td>15x15x384x200</td>
</tr>
<tr>
<td>10</td>
<td>relu</td>
<td>15x15x384x200</td>
<td>Max(0,x)</td>
<td>1</td>
<td>0</td>
<td>15x15x384x200</td>
</tr>
<tr>
<td>11</td>
<td>conv</td>
<td>15x15x384x200</td>
<td>1x1x384x384</td>
<td>1</td>
<td>0</td>
<td>15x15x384x200</td>
</tr>
<tr>
<td>12</td>
<td>relu</td>
<td>15x15x384x200</td>
<td>Max(0,x)</td>
<td>1</td>
<td>0</td>
<td>15x15x384x200</td>
</tr>
<tr>
<td>13</td>
<td>conv</td>
<td>15x15x384x200</td>
<td>1x1x384x384</td>
<td>1</td>
<td>0</td>
<td>15x15x384x200</td>
</tr>
<tr>
<td>14</td>
<td>relu</td>
<td>15x15x384x200</td>
<td>Max(0,x)</td>
<td>1</td>
<td>0</td>
<td>15x15x384x200</td>
</tr>
<tr>
<td>15</td>
<td>conv</td>
<td>15x15x384x200</td>
<td>1x1x384x384</td>
<td>1</td>
<td>0</td>
<td>15x15x384x200</td>
</tr>
<tr>
<td>16</td>
<td>relu</td>
<td>15x15x384x200</td>
<td>Max(0,x)</td>
<td>1</td>
<td>0</td>
<td>15x15x384x200</td>
</tr>
<tr>
<td>17</td>
<td>pool (avg)</td>
<td>15x15x384x200</td>
<td>3x3</td>
<td>2</td>
<td>0</td>
<td>7x7x384x200</td>
</tr>
<tr>
<td>18</td>
<td>dropout (0.55)</td>
<td>7x7x384x200</td>
<td>5x5x3</td>
<td>2</td>
<td>0</td>
<td>3x3x384x200</td>
</tr>
<tr>
<td>19</td>
<td>conv</td>
<td>7x7x384x200</td>
<td>5x5x384x384</td>
<td>1</td>
<td>2</td>
<td>7x7x384x200</td>
</tr>
<tr>
<td>20</td>
<td>relu</td>
<td>7x7x384x200</td>
<td>Max(0,x)</td>
<td>1</td>
<td>0</td>
<td>7x7x384x200</td>
</tr>
<tr>
<td>21</td>
<td>conv</td>
<td>7x7x384x200</td>
<td>1x1x384x384</td>
<td>1</td>
<td>0</td>
<td>7x7x384x200</td>
</tr>
<tr>
<td>22</td>
<td>relu</td>
<td>7x7x384x200</td>
<td>Max(0,x)</td>
<td>1</td>
<td>0</td>
<td>7x7x384x200</td>
</tr>
<tr>
<td>23</td>
<td>conv</td>
<td>7x7x384x200</td>
<td>1x1x384x384</td>
<td>1</td>
<td>0</td>
<td>7x7x384x200</td>
</tr>
<tr>
<td>24</td>
<td>relu</td>
<td>7x7x384x200</td>
<td>Max(0,x)</td>
<td>1</td>
<td>0</td>
<td>7x7x384x200</td>
</tr>
<tr>
<td>25</td>
<td>pool (max)</td>
<td>7x7x384x200</td>
<td>3x3</td>
<td>2</td>
<td>0</td>
<td>3x3x384x200</td>
</tr>
<tr>
<td>26</td>
<td>dropout (0.55)</td>
<td>3x3x384x200</td>
<td>5x5x3</td>
<td>2</td>
<td>0</td>
<td>3x3x384x200</td>
</tr>
<tr>
<td>27</td>
<td>conv</td>
<td>3x3x384x200</td>
<td>3x3x384x384</td>
<td>1</td>
<td>1</td>
<td>3x3x384x200</td>
</tr>
<tr>
<td>28</td>
<td>relu</td>
<td>3x3x384x200</td>
<td>Max(0,x)</td>
<td>1</td>
<td>0</td>
<td>3x3x384x200</td>
</tr>
<tr>
<td>29</td>
<td>conv</td>
<td>3x3x384x200</td>
<td>1x1x384x384</td>
<td>1</td>
<td>0</td>
<td>3x3x384x200</td>
</tr>
<tr>
<td>30</td>
<td>relu</td>
<td>3x3x384x200</td>
<td>Max(0,x)</td>
<td>1</td>
<td>0</td>
<td>3x3x384x200</td>
</tr>
<tr>
<td>31</td>
<td>conv</td>
<td>3x3x384x200</td>
<td>1x1x384x384</td>
<td>1</td>
<td>0</td>
<td>3x3x384x200</td>
</tr>
<tr>
<td>32</td>
<td>pool (avg)</td>
<td>3x3x200x200</td>
<td>3x3</td>
<td>1</td>
<td>0</td>
<td>1x1x200x200</td>
</tr>
<tr>
<td>33</td>
<td>softmax loss</td>
<td>1x1x200x200</td>
<td>1x1</td>
<td>0</td>
<td>1x1</td>
<td>1x1x200x200</td>
</tr>
</tbody>
</table>
Today’s class

• Review of important concepts

• Some important open problems

• Feedback and course evaluation
Fundamentals of Computer Vision

• Light
  – What an image records

• Geometry
  – How to relate world coordinates and image coordinates

• Matching
  – How to measure the similarity of two regions

• Alignment
  – How to align points/patches
  – How to recover transformation parameters based on matched points

• Grouping
  – What points/regions/lines belong together?

• Categorization
  – What similarities are important?
Light and Color

• Shading of diffuse materials depends on albedo and orientation wrt light
  – Gradients are a major cue for changes in orientation (shape)

• Many materials have a specular component that directly reflects light

• Reflected color depends on albedo and light color

• RGB is default color space, but sometimes others (e.g., HSV, L*a*b) are more useful

Image from Koenderink
Geometry

- \( \mathbf{x} = \mathbf{K} [\mathbf{R} \ \mathbf{t}] \mathbf{X} \)
  - Maps 3d point \( \mathbf{X} \) to 2d point \( \mathbf{x} \)
  - Rotation \( \mathbf{R} \) and translation \( \mathbf{t} \) map into camera’s 3D coordinates
  - Intrinsic matrix \( \mathbf{K} \) projects from 3D to 2D

- Parallel lines in 3D converge at the **vanishing point** in the image
  - A 3D plane has a vanishing line in the image

- \( \mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \)
  - Points in two views that correspond to the same 3D point are related by the fundamental matrix \( \mathbf{F} \)
Matching

• Does this patch match that patch?
  – In two simultaneous views? (stereo)
  – In two successive frames? (tracking, flow, SFM)
  – In two pictures of the same object? (recognition)
Matching

**Representation:** be invariant/robust to expected deformations but nothing else

- Assume that shape does not change
  - Key cue: local differences in shading (e.g., gradients)
- Change in viewpoint
  - Rotation invariance: rotate and/or affine warp patch according to dominant orientations
- Change in lighting or camera gain
  - Average intensity invariance: oriented gradient-based matching
  - Contrast invariance: normalize gradients by magnitude
- Small translations
  - Translation robustness: histograms over small regions

But can one representation do all of this?

- SIFT: local normalized histograms of oriented gradients provides robustness to in-plane orientation, lighting, contrast, translation
- HOG: like SIFT but does not rotate to dominant orientation
Alignment of points

Search: efficiently align matching patches

• Interest points: find repeatable, distinctive points
  – Long-range matching: e.g., wide baseline stereo, panoramas, object instance recognition
  – Harris: points with strong gradients in orthogonal directions (e.g., corners) are precisely repeatable in x-y
  – Difference of Gaussian: points with peak response in Laplacian image pyramid are somewhat repeatable in x-y-scale

• Local search
  – Short range matching: e.g., tracking, optical flow
  – Gradient descent on patch SSD, often with image pyramid

• Window/scan search
  – Long-range matching: e.g., recognition, stereo w/ scanline
Alignment of sets

Find transformation to align matching sets of points

- Geometric transformation (e.g., affine)
  - Least squares fit (SVD), if all matches can be trusted
  - Hough transform: each potential match votes for a range of parameters
    - Works well if there are very few parameters (3-4)
  - RANSAC: repeatedly sample potential matches, compute parameters, and check for inliers
    - Works well if fraction of inliers is high and few parameters (4-8)

- Other cases
  - Thin plate spline for more general distortions
  - One-to-one correspondence (Hungarian algorithm)
Grouping

• Clustering: group items (patches, pixels, lines, etc.) that have similar appearance
  – Uses: discretize continuous values; improve efficiency; summarize data
  – Algorithms: k-means, agglomerative

• Segmentation: group pixels into regions of coherent color, texture, motion, and/or label
  – Mean-shift clustering
  – Watershed
  – Graph-based segmentation: e.g., MRF and graph cuts

• EM, mixture models: probabilistically group items that are likely to be drawn from the same distribution, while estimating the distributions’ parameters
Recognition

Match objects, parts, or scenes that may vary in appearance

• Categories are typically defined by human and may be related by function, cost, or other non-visual attributes

• Key problem: what are important similarities?
  – Can be learned from training examples
Recognition

• Major improvements in feature learning (CNNs) over past five years
  – If you have lots (100K+) examples, you can learn features from scratch
  – Otherwise, use outputs from a pre-trained network

• Similar networks can be used for object detection, pose estimation, and segmentation
**Vision as part of an intelligent system**

<table>
<thead>
<tr>
<th>3D Scene</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Extraction</td>
</tr>
<tr>
<td>Texture</td>
</tr>
<tr>
<td>Grouping</td>
</tr>
<tr>
<td>Surfaces</td>
</tr>
<tr>
<td>Interpretation</td>
</tr>
<tr>
<td>Objects</td>
</tr>
<tr>
<td>Action</td>
</tr>
<tr>
<td>Walk, touch, contemplate, smile, evade, read on, pick up, ...</td>
</tr>
</tbody>
</table>
Well-Established
(patch matching)

Face Detection/Recognition

Object Tracking / Flow

Multi-view Geometry

Major Progress
(pattern matching++)

Category Detection

Human Pose

3D Scene Layout

New Opportunities
(translation/tasks)

Entailment/Prediction

Life-long Learning

Vision for Robots
Vision is ready to break out of the lab

Vision is not the goal, but a way to learn about your immediate environment and the world, so that you can act.
Scene Understanding =
Objects + People + Layout +
Interpretation *within Task Context*
What do I see?  →  Why is this happening?
What is important?
What will I see?

How can we learn about the world through vision?

How do we create/evaluate vision systems that adapt to useful tasks?
Important open problems

• How can we interpret vision given structured plans of a scene?
Important open problems

- Algorithms: works pretty well → nearly perfect
  - Stereo: top of wish list from Pixar guy Micheal Kass
  - Structure-from-motion

Good directions:
- Incorporate higher level knowledge
Important open problems

• Spatial understanding

Important questions:

• What are good representations of space for navigation and interaction? What kind of details are important?
• How can we combine single-image cues with multi-view cues?
Important open problems

Object representation: what is it?

Important questions:
• How can we pose recognition so that it lets us deal with new objects?
• What do we want to predict or infer, and to what extent does that rely on categorization?
• How do we transfer knowledge of one type of object to another?
Important open problems

• Can we build a “core” vision system that can easily be extended to perform new tasks or even learn on its own?
  – What kind of representations might allow this?
  – What should be built in and what should be learned?
Important open problems

• Vision for the masses

Counting cells

Analyzing social effects of green space

How to make vision systems that can quickly adapt to these thousands of visual tasks?
If you want to learn more...

• Read lots of papers: IJCV, PAMI, CVPR, ICCV, ECCV, NIPS

• Helpful topics for classes
  – David Forsyth’s optimization
  – Classes in machine learning or pattern recognition
  – Statistics, graphical models
  – Seminar-style paper-reading classes

• Just implement stuff, try demos, see what works
Feedback through Google Form

More specific feedback to help improve course

https://goo.gl/forms/FIfUqnVH8pcGHoCe2
ICES Forms

• Looking forward to your project presentations!