Face Recognition
Therefore, when our goal is to recognize people from their face images, we do not have to determine the location of these features.
Instead, we can normalize the face images as best as possible (for example, make sure that the eyes are always in the same two image positions) and then compute descriptors for fixed sections of the images.
The most popular and successful descriptor of this kind is referred to as **Local Binary Patterns (LBPs)**.

Local Binary Patterns
For each pixel in a given section, we look at its 8 neighbors (we can also pick neighbors at radii greater than 1; radius 2 is most popular).
Then we form a binary string by looking at each neighbor in a specified order and adding a "1" to a binary string if it is brighter than the center pixel and "0" otherwise.
The binary string can be interpreted as a number.

Local Binary Patterns
For a given area of pixels, we can now compute how many fall into each of these 58 categories or the "inconsistent" category.
This results in a histogram with 59 bins.
We can split a given face image into, for example, 7x6 sections and obtain a histogram for each section.
Then we concatenate these histograms to obtain a feature vector that describes the given face image.
In our example, these feature vectors have $7 \times 6 \times 59 = 2478$ elements.

Local Binary Patterns
Different patterns (numbers) describe different types of gradients, i.e., local changes in intensity.
A consistent gradient should result in exactly one uninterrupted sequence of 1s and the remaining bits should be 0s, as in the 5 examples below:
There are exactly 58 patterns of this kind.

Local Binary Patterns
Some of these sections are more important for recognition than others, so we may assign weights:
For LBP descriptors $A_{ij}$ and $B_{ij}$ and weights $w_{ij}$, with bins $i$ and sections $j$, a good similarity measure is:
$$
\chi^2 = \sum_{i,j} w_{ij} \frac{(A_{ij} - B_{ij})^2}{A_{ij} + B_{ij}}
$$
Convolutional Neural Networks (CNNs)

Convolutional networks perform convolution to extract local features instead of having complete connectivity between layers. They also employ pooling layers to make classification more robust towards slight changes in feature locations.

Common Classification Tasks

- **Recognition of individual objects/faces**
  - Analyze object-specific features (e.g., key points)
  - Train with images from different viewing angles
- **Recognition of object classes**
  - Analyze features that are consistent within class and differ between them as much as possible.
  - Train with many exemplars from each class.
- **Recognition of scene types**
  - Find and analyze common features, objects, or layouts within scene classes.
  - Use large variety of scene photos.

Example: AlexNet

*AlexNet (Krizhevsky et al. 2012)*

The class with the highest likelihood is the one the DNN selects

When AlexNet is processing an image, this is what is happening at each layer.

Another Example: Inception

*9 Inception modules*

Network in a network in a network...

Visual Illusions

Visual Illusions demonstrate how we perceive an “interpreted version” of the incoming light pattern rather than the exact pattern itself.
Visual Illusions

He we see that the squares A and B from the previous image actually have the same luminance (but in their visual context are interpreted differently).

December 6, 2018 Introduction to Artificial Intelligence
Lecture 24: Computer Vision IV

Visual Illusions

• The purpose of biological vision is to detect relevant patterns in the environment (e.g., danger, food).

• Our visual system uses both low- and high-level information to account for changes in lighting or perspective.

• Based on its experience, it decides that the figure most likely shows a checkerboard with a shadow cast onto it.

• According to that interpretation, the actual squares must differ in brightness.

• For biological vision, the ability to recognize objects and materials under different conditions is more important than being able to determine the absolute brightness of a surface.

The End