

Images search

Stanislav Protasov

Agenda

- How our eyes work
- Historical approach to images search
- Duplicates search and CBIR
- Image and video understanding

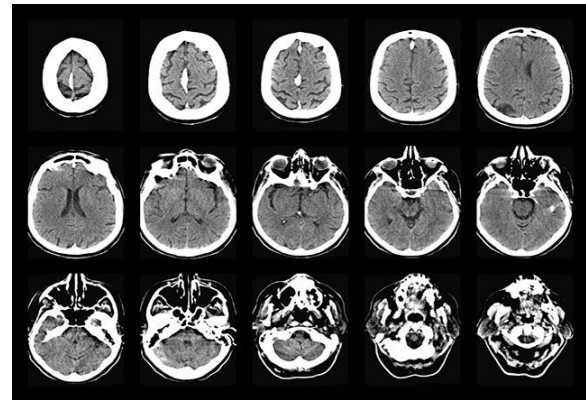
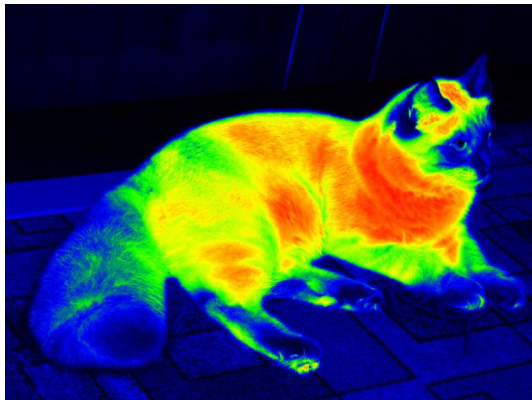
How our vision works

Hint: *very similar to digital camera*

Vision

Vision is a sensor system, that receives information using **electromagnetic waves** [of visible spectrum].

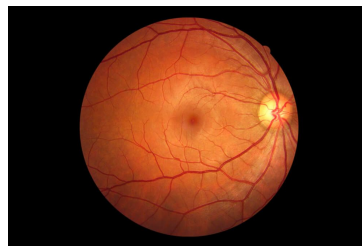
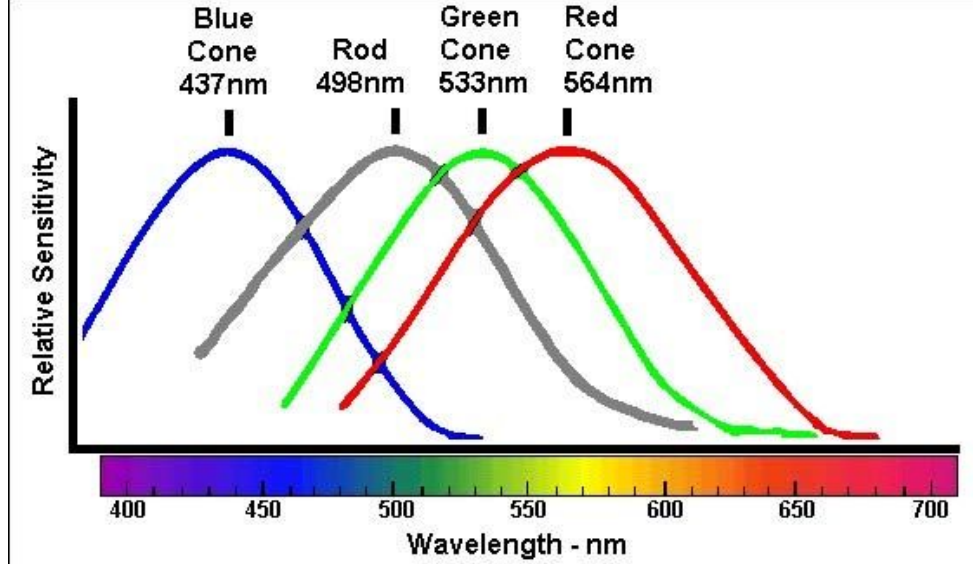
In general, X-ray, infrared and CT can be considered as “vision”.

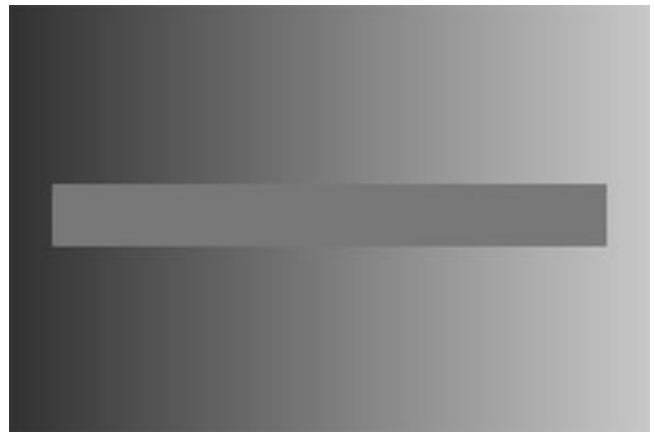
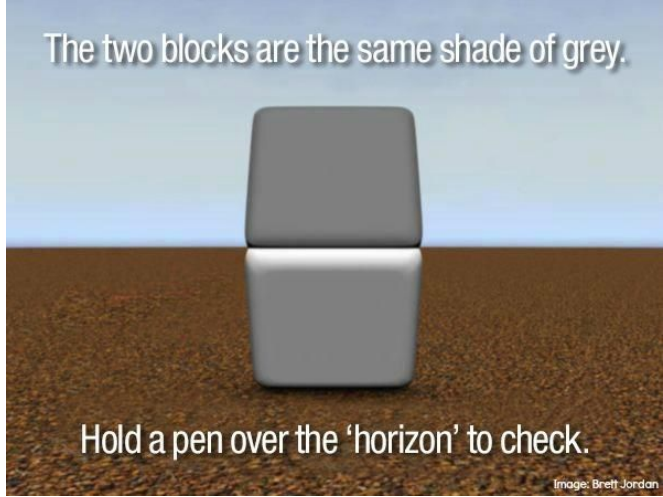
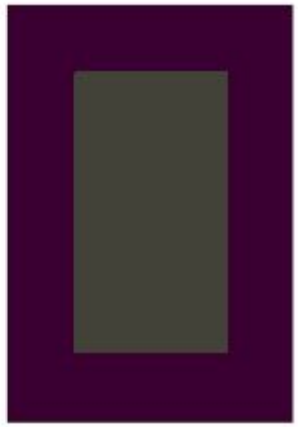
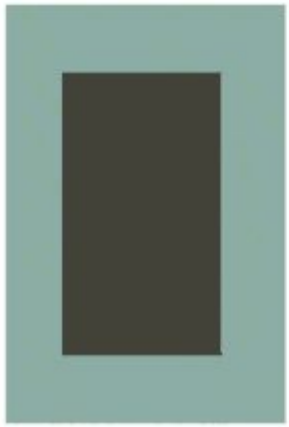
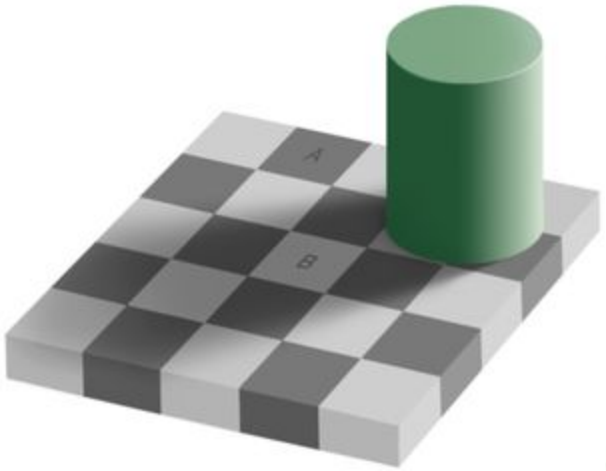


Human vision

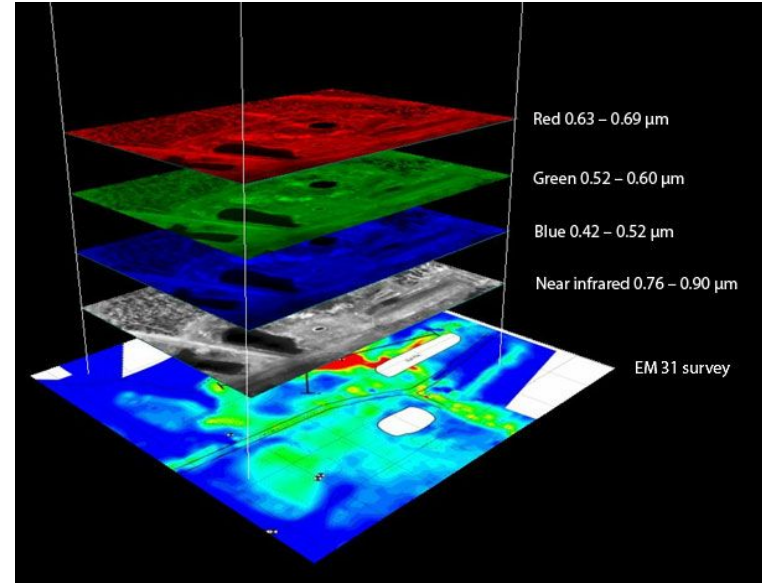
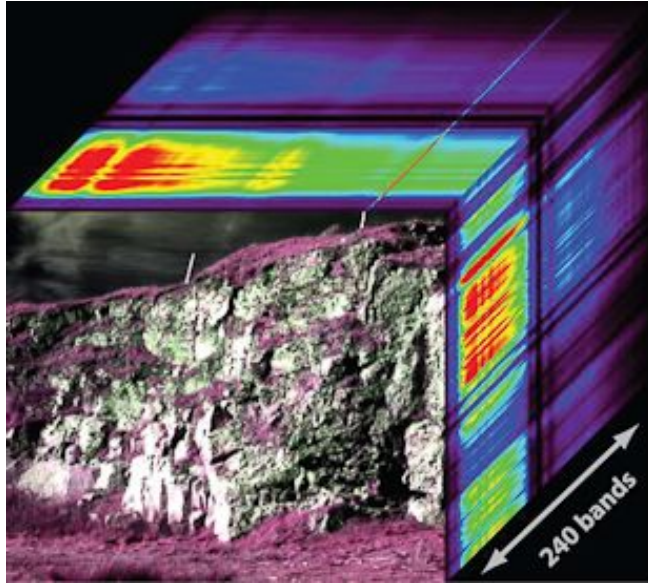
Major facts about vision:

- **Binocular** — allows restore 3D
- **Retina** — discrete
- **Color** — quantized
 - 4 types of sensor cells:
 - S,M,L-cone cells
 - Rod cells
- **Polarization and phase insensitive**
- Supports **focus**
- **Opponent-process** theory and
- **Color constancy**
 - Brain process differences of colors





Multi- and hyperspectral images



What is digital image

Digital image is a *quantized* and *discrete* vector field (similar to human vision). Each vector component describes:

- How much **energy is reflected** in particular spectrum part
 - Images, infrared images, ...

OR

- How much **energy is absorbed**
 - Medical imaging (X-ray, CT)

How images are (were) retrieved

Neighbouring text and subtitles



```
<div class="thumbinner" style="width:222px;">
  <a href="/wiki/File:Common_Hoopoe_(Upupa_epops)_at_Hodal_I_IMG_9225.jpg" class=
    "image">
     == $0
    </a>
  <div class="thumbcaption">
    <div class="magnify">...</div>
    "The muscles of the head allow the hoopoe's bill to be opened when it is
      inserted into the ground"
    </div>
  </div>
```

the male.^[4]

High-level features for filtering

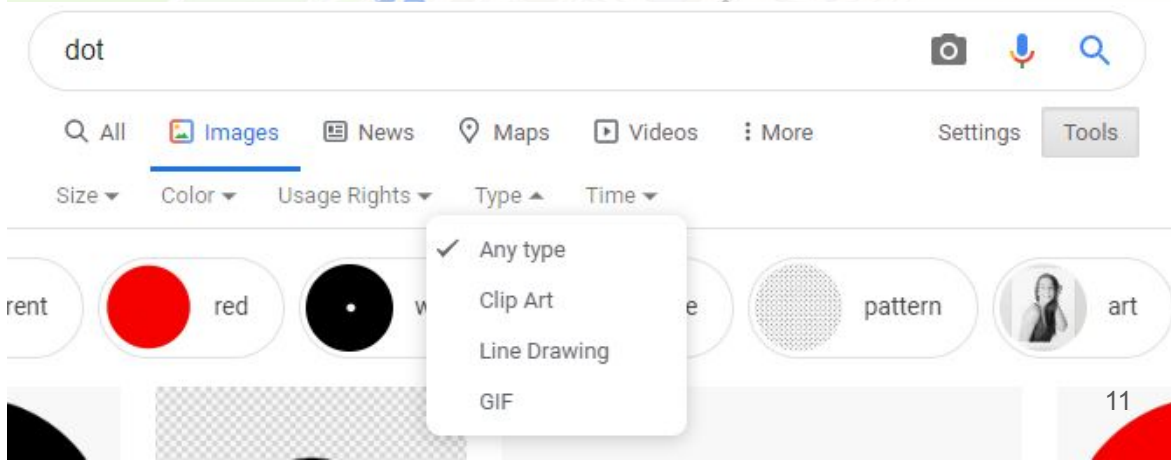
- Color (using k-Means clustering)
- Textures (Haralick/GLCM features, wavelets), shapes and easily computable features (drawings vs photos, ...)
- Metadata (size, EXIF metadata)

Поиск Картинки Видео Карты Маркет Новости Пере

Размер ▾ Ориентация ▾ Тип ^ Цвет ▾ Файл ▾ Товары

По вашему запросу ничего не найдено

- ☒ Любой тип
- ☐ Фото
- ☐ С белым фоном
- ☐ Рисунки и чертежи
- ☐ Лица
- ☐ Демотиваторы



CBIR = Content Based Image Retrieval

CBIR

Problems (sensitivity increases)

- Similarity search
- Duplicate search
- Identification (exactly the same, but with respect to e.g. compression)

Similarity and duplicate search: image as a *bag of words*

In CV ... a **feature [point]** is defined as an "interesting" part of an image.

Usually for **interesting points** consider:

- Edges
- Corners
- Regions

After detector *feature vector* (**descriptor**) is computed.

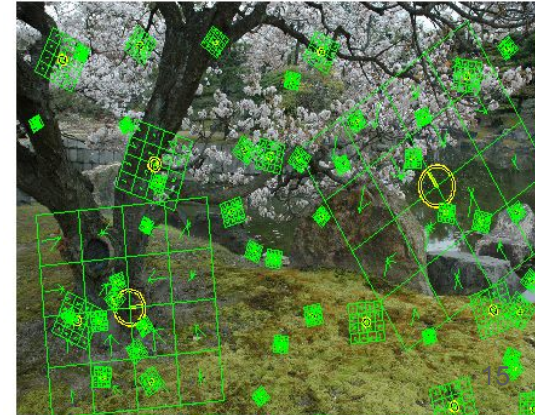
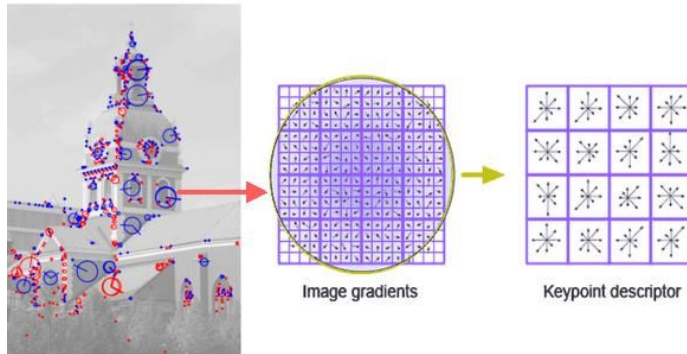
Use feature vector sets to describe **objects**



$$\Delta[G_{\sigma}(x, y) * f(x, y)] = [\Delta G_{\sigma}(x, y)] * f(x, y) = LoG * f(x, y)$$

SIFT: Scale-invariant feature transform

- 1) Compute gradients for images in *image pyramid* using difference of Gaussians (DoG). (Image pyramid ~ Scale invariant)
- 2) Search for local extrema in scale and space (*keypoints*)
- 3) Compute *direction* (*rotation invariant*)
- 4) Create descriptor: in 16x16 neighbourhood make 16 blocks, compute gradients (8 bins for angles) and make a vector.
- 5) Normalize (*intensity invariant*)



SIFT

overview

Invariant Local Features

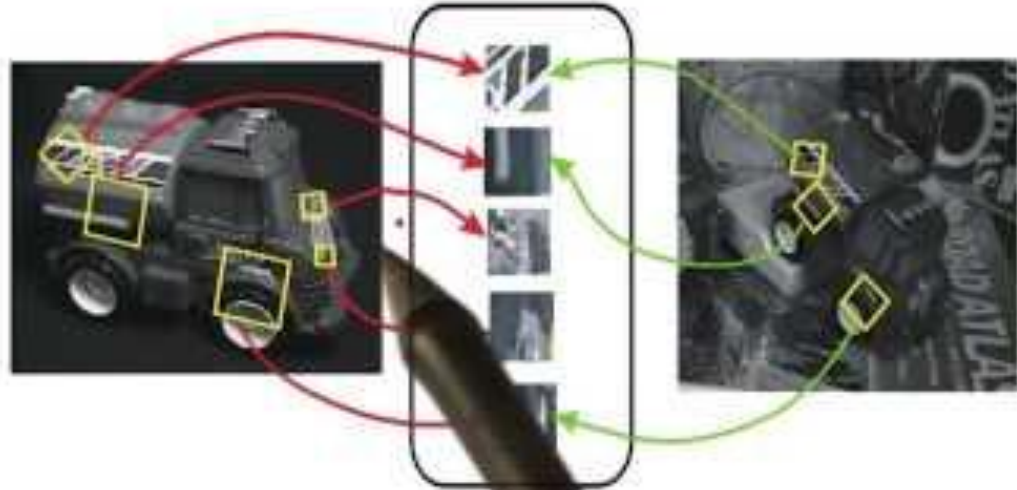
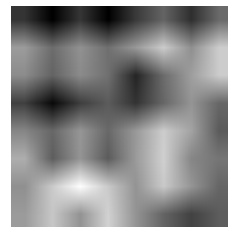


Image fingerprinting for **duplicate search**

1. Use PoI. Allows cropping, need ~100 points, fails for texts
2. Use hash functions:
 - a. Image.Match based on Xerox features
 - Grayscale color image
 - Place 9x9 uniform grid of pixels
 - Each point is described with 8-neighbourhood {darker = -2, mild darker , ... , lighter = +2 }
 - Concatenate

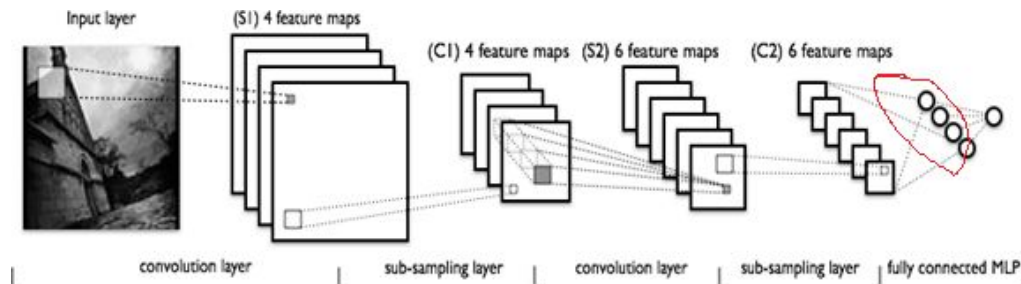
Image fingerprinting for duplicate search (2)

- Hash functions ([pip install ImageHash](#)):
- [\[average\] aHash](#)
 - Resize to 8x8
 - Grayscale
 - Binarize by average
 - Use Hamming dist
- [\[perception\] pHash](#) and [\[wavelet\] wHash](#)
 - pHash uses DCT
 - wHash - DWT, both coarse grained
 - Use Hamming dist
- [\[difference\] dHash](#)
 - Resize to 9x8
 - Grayscale
 - Compute $I[x+1, y] <> I[x, y]$ and use this as a bit



Deep networks for specific and general **similarity** search

1. Images are of **different types** (classes, e.g. ImageNet). [Train classification network](#) (AlexNet, VGG16, ...) and use embeddings (from inner layer) as index.



2. Images are of the same type (faces). Train deep [convolutional autoencoder](#) which creates small-dimensional embeddings.

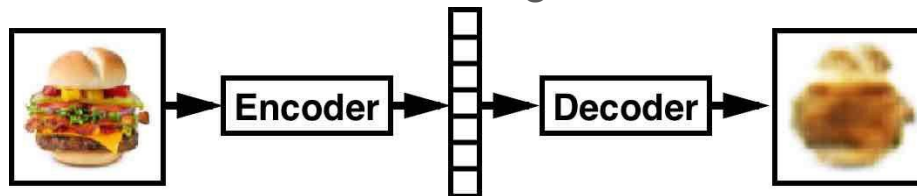


Image understanding, video structure

Semantic retrieval

Deep classification and region-based networks allow adding semantic indices.

NB:

- *How many \$\$ will single inference cost for 20B of images?*
- *How much time?*
- *How often should a company do it?*

Video structure mining

As text can be searched for a **paragraph**,
Long videos should be also indexed with
scenes. [[demo](#)]

