EECS 4422/5323 Computer Vision

Feature Detection Lecture 1

Calden Wloka

18 September, 2019

Announcments

- Readings are shifted - we'll cover Section 4.2 next class

Outline

- Finish Image Representation
  - Connected Components
  - Frequency Domain
- Introduction to Features
- Project Ideas

Median Filter Demo

Go to median filter demo.

Blobs

Last class we ended on a discussion of morphology over binary maps (*e.g.* erosion and dilation). Those techniques are useful for binary map manipulation, but we often want to automatically find clusters over a binary map.

- These clusters are often referred to as "blobs" or "connected components"

Blobs

Last class we ended on a discussion of morphology over binary maps (*e.g.* erosion and dilation). Those techniques are useful for binary map manipulation, but we often want to automatically find clusters over a binary map.

- These clusters are often referred to as "blobs" or "connected components"
- Calculated using common techniques in graph theory

Blobs

Last class we ended on a discussion of morphology over binary maps (*e.g.* erosion and dilation). Those techniques are useful for binary map manipulation, but we often want to automatically find clusters over a binary map.

- These clusters are often referred to as “blobs” or “connected components”
- Calculated using common techniques in graph theory
- Can be 4-connected (horizontal or vertical neighbour only) or 8-connected (diagonal neighbours too)

Blobs

Last class we ended on a discussion of morphology over binary maps (*e.g.* erosion and dilation). Those techniques are useful for binary map manipulation, but we often want to automatically find clusters over a binary map.

- These clusters are often referred to as "blobs" or "connected components"
- Calculated using common techniques in graph theory
- Can be 4-connected (horizontal or vertical neighbour only) or 8-connected (diagonal neighbours too)
- Commonly associated with a number of attributes:

Blobs

Last class we ended on a discussion of morphology over binary maps (*e.g.* erosion and dilation). Those techniques are useful for binary map manipulation, but we often want to automatically find clusters over a binary map.

- These clusters are often referred to as "blobs" or "connected components"
- Calculated using common techniques in graph theory
- Can be 4-connected (horizontal or vertical neighbour only) or 8-connected (diagonal neighbours too)
- Commonly associated with a number of attributes:
    - Centroid

Blobs

Last class we ended on a discussion of morphology over binary maps (*e.g.* erosion and dilation). Those techniques are useful for binary map manipulation, but we often want to automatically find clusters over a binary map.

- These clusters are often referred to as "blobs" or "connected components"
- Calculated using common techniques in graph theory
- Can be 4-connected (horizontal or vertical neighbour only) or 8-connected (diagonal neighbours too)
- Commonly associated with a number of attributes:
  - Centroid
  - Area

Blobs

Last class we ended on a discussion of morphology over binary maps (*e.g.* erosion and dilation). Those techniques are useful for binary map manipulation, but we often want to automatically find clusters over a binary map.

- These clusters are often referred to as “blobs” or “connected components”
- Calculated using common techniques in graph theory
- Can be 4-connected (horizontal or vertical neighbour only) or 8-connected (diagonal neighbours too)
- Commonly associated with a number of attributes:
    - Centroid
    - Area
    - Bounding Box

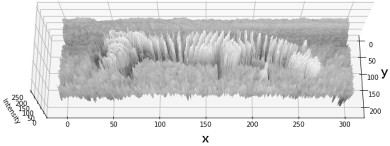
Connected Components in Action

We can use connected components to help clean up noise in our binary map.

Go to updated α-matte with morphology demo.

Reminder: Images as Signals

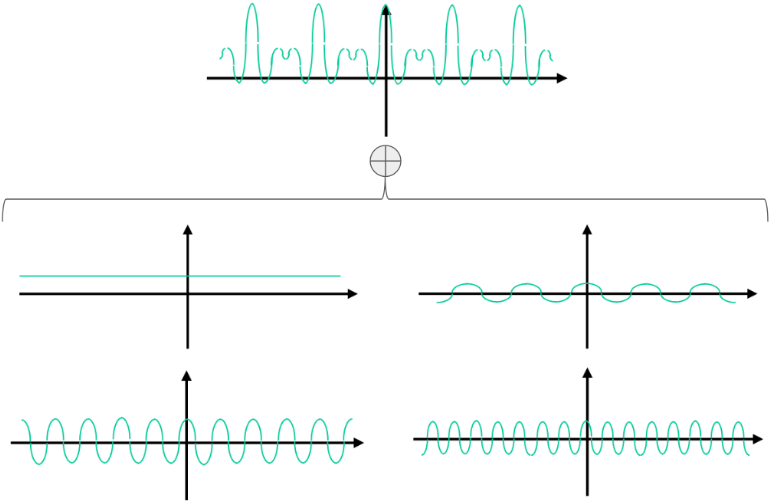
Image



Manifold visualization

The Basics of Signal Analysis

Arbitrary signals can be decomposed into a sum of component signals.



Adapted from Richard Wildes' lecture slides.

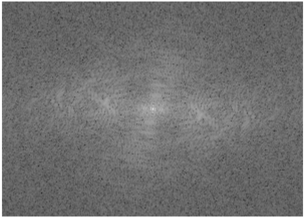
The Basics of Signal Analysis

A helpful video visualization can be found here.

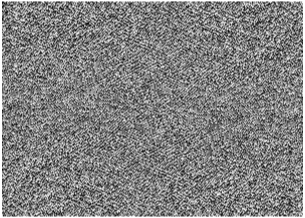
Frequency Representations of Images

We can decompose an image into its spectral components.

Image



Magnitude

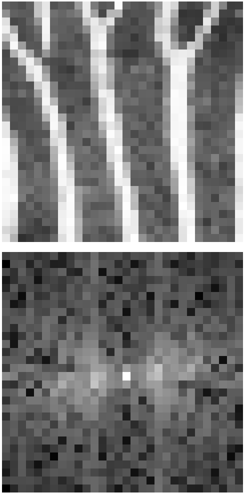


Phase

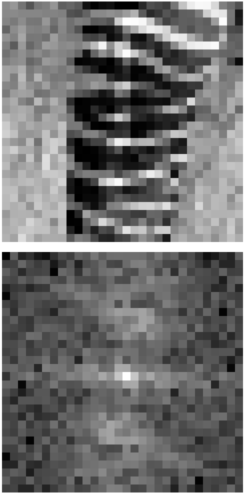
Convolutions, particularly band-pass filtering, can be very quickly calculated in the frequency domain

Patch Examples

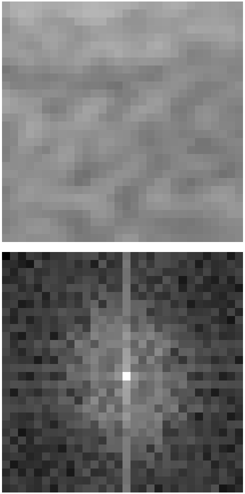
Body Patch



Foreleg Patch



Background Patch



Some Comments About Frequency Representation

- The value depends on balancing the cost of transforming to a frequency representation (FFT) and the improved efficiency of calculating convolutions

Some Comments About Frequency Representation

- The value depends on balancing the cost of transforming to a frequency representation (FFT) and the improved efficiency of calculating convolutions
- FFT conversions often happen "under the hood"

Some Comments About Frequency Representation

- The value depends on balancing the cost of transforming to a frequency representation (FFT) and the improved efficiency of calculating convolutions
- FFT conversions often happen “under the hood”
- We will predominantly discuss filtering and show examples in the spatial domain, but know that it could equivalently be done in the frequency domain

Some Comments About Frequency Representation

- The value depends on balancing the cost of transforming to a frequency representation (FFT) and the improved efficiency of calculating convolutions
- FFT conversions often happen "under the hood"
- We will predominantly discuss filtering and show examples in the spatial domain, but know that it could equivalently be done in the frequency domain
- Frequency domain calculations are the basis for frequently used vocabulary when discussing image features (*e.g.* "High-frequency" and "Low-frequency")

Features are Informative Attributes

A *feature* is a flexible concept representing information which is useful.

- Features can be assigned at the pixel level, essentially creating a new information channel (*e.g.* α channel, edge images)

Features are Informative Attributes

A *feature* is a flexible concept representing information which is useful.

- Features can be assigned at the pixel level, essentially creating a new information channel (*e.g.* α channel, edge images)
- Features can be assigned at the image level (*e.g.* Does the image contain X, benchmark scores)

Features are Informative Attributes

A *feature* is a flexible concept representing information which is useful.

- Features can be assigned at the pixel level, essentially creating a new information channel (*e.g.* α channel, edge images)
- Features can be assigned at the image level (*e.g.* Does the image contain X, benchmark scores)
- Features may be assigned sparsely or anisotropically (*e.g.* Blob centroids)

Features are Informative Attributes

A *feature* is a flexible concept representing information which is useful.

- Features can be assigned at the pixel level, essentially creating a new information channel (*e.g.* α channel, edge images)
- Features can be assigned at the image level (*e.g.* Does the image contain X, benchmark scores)
- Features may be assigned sparsely or anisotropically (*e.g.* Blob centroids)
- Features can interact with other features

Features are Informative Attributes

A *feature* is a flexible concept representing information which is useful.

- Features can be assigned at the pixel level, essentially creating a new information channel (*e.g.* α channel, edge images)
- Features can be assigned at the image level (*e.g.* Does the image contain X, benchmark scores)
- Features may be assigned sparsely or anisotropically (*e.g.* Blob centroids)
- Features can interact with other features
    - Hierarchically (*e.g.* Convolutional Neural Networks)

Features are Informative Attributes

A *feature* is a flexible concept representing information which is useful.

- Features can be assigned at the pixel level, essentially creating a new information channel (*e.g.* α channel, edge images)
- Features can be assigned at the image level (*e.g.* Does the image contain X, benchmark scores)
- Features may be assigned sparsely or anisotropically (*e.g.* Blob centroids)
- Features can interact with other features
  - Hierarchically (*e.g.* Convolutional Neural Networks)
  - (Semi) Independently (*e.g.* Support Vector Machines)

Feature Vectors

A *feature vector* is an ordered set of feature values.

Examples include:

- Lowe, Object Recognition from Local Scale-Invariant Features (SIFT)
  128 element feature vector based on local histogram descriptors of image keypoints
- Judd *et al.*, Learning to Predict Where Humans Look
  33 features concatenated into a vector, including steerable bands, colour and luminance channels, distance to the image center, and the output of a horizon line detector, face detector, and person detector

Feature Spaces

Taking the possible range over all elements of a feature vector produces a *feature space*. Any given instantiation of a feature vector defines a point within that space.

Feature Spaces

Taking the possible range over all elements of a feature vector produces a *feature space*. Any given instantiation of a feature vector defines a point within that space.

Two major challenges when dealing with feature spaces is that they frequently have a very high dimensionality and a highly heterogeneous spread of data.

Feature Spaces

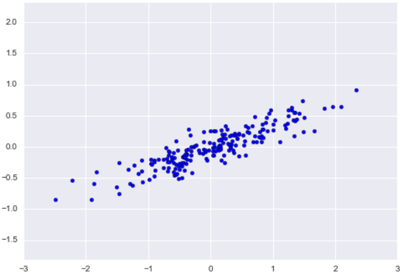
Taking the possible range over all elements of a feature vector produces a *feature space*. Any given instantiation of a feature vector defines a point within that space.

Two major challenges when dealing with feature spaces is that they frequently have a very high dimensionality and a highly heterogeneous spread of data.

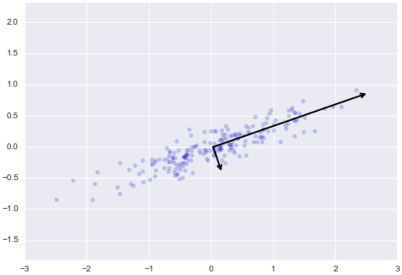
This may or may not be a problem. We may still be able to either learn a mapping from our vector space to a desirable output space (or alternative feature space) or directly reason over our features. Sometimes, however, it is necessary or useful to try and re-orient our data to a more optimal set of axes, and, particularly for visualization, to reduce the dimensionality.

Principal Component Analysis (PCA)

PCA is a highly useful technique for dimensionality reduction, but it can also be used for other tasks (like better controlling the variance across feature dimensions).



Data



Principal Components

For a comprehensive look at using and interpreting PCA, check out this work by VanderPlas.

A Clerical Note

- The white paper doesn't have to be the first time we've discussed your project idea!

I've spoken with many of you and mentioned email but haven't gotten much mail... so, just for the record, you can reach me at: calden[at]eecs.yorku.ca

A Clerical Note

- The white paper doesn't have to be the first time we've discussed your project idea!
- Email provides a useful record over verbal communication

I've spoken with many of you and mentioned email but haven't gotten much mail... so, just for the record, you can reach me at: calden[at]eecs.yorku.ca

A Clerical Note

- The white paper doesn't have to be the first time we've discussed your project idea!
- Email provides a useful record over verbal communication
- The discussion can be iterative; you don't need to know what all parts of your project will look like to know what you want some parts to look like

I've spoken with many of you and mentioned email but haven't gotten much mail... so, just for the record, you can reach me at: calden[at]eecs.yorku.ca

Scientific Projects

- Science is about the discovery of new knowledge or new understandings

Scientific Projects

- Science is about the discovery of new knowledge or new understandings
- Computer vision now is highly dominated by engineering approaches, but there is still room for scientific exploration

Scientific Projects

- Science is about the discovery of new knowledge or new understandings
- Computer vision now is highly dominated by engineering approaches, but there is still room for scientific exploration
- Question driven

Scientific Projects

- Science is about the discovery of new knowledge or new understandings
- Computer vision now is highly dominated by engineering approaches, but there is still room for scientific exploration
- Question driven
  - What are you testing? What do you expect to find?

Scientific Projects

- Science is about the discovery of new knowledge or new understandings
- Computer vision now is highly dominated by engineering approaches, but there is still room for scientific exploration
- Question driven
  - What are you testing? What do you expect to find?
  - Why is this a useful avenue of exploration?

A Review of Some Suggested Approaches

We won't go through all suggested projects; the goal here is to highlight a few and provide ideas and examples of potential approaches.

- Image Preprocessing
- Module Combination
- Algorithm Comparisons
- Principled Probing of Algorithm Behaviour

Image Preprocessing

- This branch of projects looks at how manipulating the input to a computer vision model(s) affects its performance or behaviour

Image Preprocessing

- This branch of projects looks at how manipulating the input to a computer vision model(s) affects its performance or behaviour
- You have already mastered many of the necessary skills for this branch of projects

Image Preprocessing

- This branch of projects looks at how manipulating the input to a computer vision model(s) affects its performance or behaviour
- You have already mastered many of the necessary skills for this branch of projects
- The description on the website could actually be improved; this project could explore both improvement and degradation of performance

Noise and Model Robustness

Image noise is any sort of erroneous or aberrant component of the visual signal which obscures the underlying reality or disrupts our ability to extract useful information.

- Disclaimer: Not all pre-processing projects need to revolve around noise, it's just a suggested starting point!

Noise and Model Robustness

Image noise is any sort of erroneous or aberrant component of the visual signal which obscures the underlying reality or disrupts our ability to extract useful information.

- Disclaimer: Not all pre-processing projects need to revolve around noise, it's just a suggested starting point!
- There are a number of common noise models which could be applied independantly or jointly

Noise and Model Robustness

Image noise is any sort of erroneous or aberrant component of the visual signal which obscures the underlying reality or disrupts our ability to extract useful information.

- Disclaimer: Not all pre-processing projects need to revolve around noise, it's just a suggested starting point!
- There are a number of common noise models which could be applied independantly or jointly
- Many computer vision models are tested on "good" data, which opens a couple avenues of exploration

Noise and Model Robustness

Image noise is any sort of erroneous or aberrant component of the visual signal which obscures the underlying reality or disrupts our ability to extract useful information.

- Disclaimer: Not all pre-processing projects need to revolve around noise, it's just a suggested starting point!
- There are a number of common noise models which could be applied independantly or jointly
- Many computer vision models are tested on “good” data, which opens a couple avenues of exploration
  - How does the model handle different types of noise? Is it robust or does it degrade?

Noise and Model Robustness

Image noise is any sort of erroneous or aberrant component of the visual signal which obscures the underlying reality or disrupts our ability to extract useful information.

- Disclaimer: Not all pre-processing projects need to revolve around noise, it's just a suggested starting point!
- There are a number of common noise models which could be applied independantly or jointly
- Many computer vision models are tested on "good" data, which opens a couple avenues of exploration
    - How does the model handle different types of noise? Is it robust or does it degrade?
    - Do common pre-processing steps appropriate to that type of noise mitigate its effects?

Noise and Model Robustness

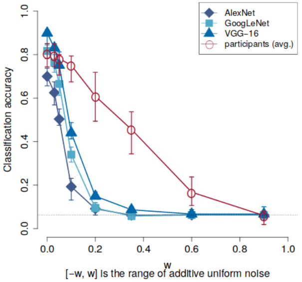
Image noise is any sort of erroneous or aberrant component of the visual signal which obscures the underlying reality or disrupts our ability to extract useful information.

- Disclaimer: Not all pre-processing projects need to revolve around noise, it's just a suggested starting point!
- There are a number of common noise models which could be applied independantly or jointly
- Many computer vision models are tested on "good" data, which opens a couple avenues of exploration
  - How does the model handle different types of noise? Is it robust or does it degrade?
  - Do common pre-processing steps appropriate to that type of noise mitigate its effects?
  - Does adding pre-processing affect performance over "good" data?

An Example from the Literature

Geirhos *et al.*, 2017, Comparing deep neural networks against humans: object recognition when the signal gets weaker, arXiv link

Geirhos *et al.* looked at the performance change for human subjects and several object recognition deep networks over a range of additive noise, and found that deep network performance degrades much faster than human performance.



(c) Noise-experiment accuracy

Plot of results from Geirhos *et al.*

Module Combination

This is a very general approach, and is quite common in computer vision. It essentially views the set of established methods in computer vision as a toolkit, and selectively combines a novel set of tools with the aim of improving performance.

Module Combination

This is a very general approach, and is quite common in computer vision. It essentially views the set of established methods in computer vision as a toolkit, and selectively combines a novel set of tools with the aim of improving performance.

In some ways, the “primitive event recognition” suggestion is a subset of module combination, in that the first module is the primitive event detector, and the second module is whatever method is applied over the pruned feature set.

Example Method for Feature Weighting: Saliency

A *saliency map* is a computational model which attempts to output a spatiotopic map of scene conspicuity, where each pixel is assigned a *saliency value* which corresponds to how interesting or important that pixel is.

Input image

Saliency map (IKN)

Example Method for Feature Weighting: Saliency

A *saliency map* is a computational model which attempts to output a spatiotopic map of scene conspicuity, where each pixel is assigned a *saliency value* which corresponds to how interesting or important that pixel is.

Input image

Saliency map (IKN)

The saliency map can either be thresholded to become a binary mask to prune features, or can provide a soft-weighting over features.

There are many approaches to saliency computation

Input image

Saliency map (IKN)

Saliency map (SALICON)

A Useful Tool for Saliency Approaches

The Saliency Model Implementation Library for Experimental Research (SMILER) provides a unified API and straightforward CLI for executing a wide variety of saliency models.

It can be found on GitHub, and a paper describing it can be found on arXiv.

An Example from the Literature

Feichtenhofer *et al.*, 2015, Dynamically encoded actions based on spate-time saliency, CVPR - PDF

Note that this example uses spatiotemporal saliency for video analysis; you could also adapt static saliency for static image analysis.

Modifying an Existing Pipeline

As mentioned, many computer vision models are already collections of sub-modules tied together into a processing pipeline. Another valid approach to this topic would be select one aspect of this pipeline for which there are a number of possible approaches and test all the different approaches to see which one is more effected (and why).

Modifying an Existing Pipeline

As mentioned, many computer vision models are already collections of sub-modules tied together into a processing pipeline. Another valid approach to this topic would be select one aspect of this pipeline for which there are a number of possible approaches and test all the different approaches to see which one is more effected (and why).

Some possible options include:

- A spatiotemporal model which incorporates optical flow. There are many different ways that optical flow may be computed.

Modifying an Existing Pipeline

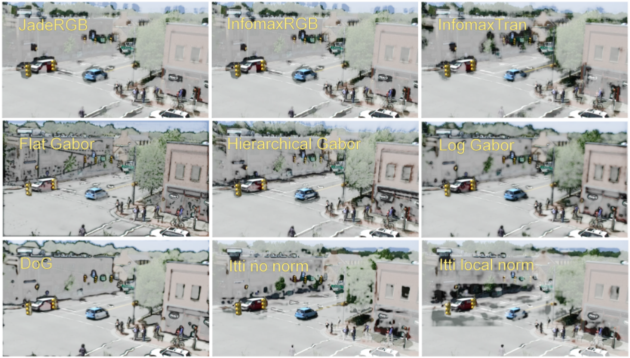
As mentioned, many computer vision models are already collections of sub-modules tied together into a processing pipeline. Another valid approach to this topic would be select one aspect of this pipeline for which there are a number of possible approaches and test all the different approaches to see which one is more effected (and why).

Some possible options include:

- A spatiotemporal model which incorporates optical flow. There are many different ways that optical flow may be computed.
- A stereoscopic model which performs feature matching between corresponding frames. There are many different models of features which could be tried.

An Example from the Literature

Bruce *et al.*, 2011, Visual Representation in the Determination of Saliency, CRV - Paper link



Example from Bruce *et al.*

Algorithm Comparison

It's not always clear which models will perform better or worse than others in a given task domain or set of inputs (a dataset). Quantifying model performance is an important avenue of research.

Algorithm Comparison

It's not always clear which models will perform better or worse than others in a given task domain or set of inputs (a dataset). Quantifying model performance is an important avenue of research.

Module Combination looked at comparing one general pipeline with different subcomponents. Algorithm comparison is doing something similar, except it is comparing multiple independent pipelines which are all aiming to produce the same type of output.

Examples from the Literature: Benchmarks

Benchmarks are large-scale comparisons of model performance which attempt to standardize model evaluation to avoid experimenter bias.

- The MIT Saliency Benchmark

Examples from the Literature: Benchmarks

Benchmarks are large-scale comparisons of model performance which attempt to standardize model evaluation to avoid experimenter bias.

- The MIT Saliency Benchmark
- The KITTI Benchmarking Suite

Examples from the Literature: Benchmarks

Benchmarks are large-scale comparisons of model performance which attempt to standardize model evaluation to avoid experimenter bias.

- The MIT Saliency Benchmark
- The KITTI Benchmarking Suite
- A Benchmark for Semantic Image Segmentation

Possible Project Options

- Find a new dataset and test performance

Possible Project Options

- Find a new dataset and test performance
  - The dataset should have ground truth labeling which is related to the expected output of the models so they can be fairly evaluated

Possible Project Options

- Find a new dataset and test performance
  - The dataset should have ground truth labeling which is related to the expected output of the models so they can be fairly evaluated
  - Some helpful resources for finding existing datasets:

Possible Project Options

- Find a new dataset and test performance
  - The dataset should have ground truth labeling which is related to the expected output of the models so they can be fairly evaluated
  - Some helpful resources for finding existing datasets:
    - VisualData, a collection of computer vision datasets

Possible Project Options

- Find a new dataset and test performance
  - The dataset should have ground truth labeling which is related to the expected output of the models so they can be fairly evaluated
  - Some helpful resources for finding existing datasets:
    - VisualData, a collection of computer vision datasets
    - CVonline, another collection

Possible Project Options

- Find a new dataset and test performance
  - The dataset should have ground truth labeling which is related to the expected output of the models so they can be fairly evaluated
  - Some helpful resources for finding existing datasets:
    - VisualData, a collection of computer vision datasets
    - CVonline, another collection
- Take an established benchmark or dataset and modify it (*e.g.* add noise) to see how that affects performance

Possible Project Options

- Find a new dataset and test performance
  - The dataset should have ground truth labeling which is related to the expected output of the models so they can be fairly evaluated
  - Some helpful resources for finding existing datasets:
    - VisualData, a collection of computer vision datasets
    - CVonline, another collection
- Take an established benchmark or dataset and modify it (*e.g.* add noise) to see how that affects performance
- Treat algorithm performance as features over the images, and see if you can identify patterns in the performance of different models

Principled Probing

Principled probing is really a subset of algorithm comparison, except that the emphasis is on selecting a specific set of images which are designed to identify weak points in algorithm design or performance.

Principled Probing

Principled probing is really a subset of algorithm comparison, except that the emphasis is on selecting a specific set of images which are designed to identify weak points in algorithm design or performance.

We saw an example of this early in this course with the discovery that CNNs typically identify sheep in landscapes without sheep and misidentify sheep in arms or indoors as dogs or cats.

Principled Probing

Principled probing is really a subset of algorithm comparison, except that the emphasis is on selecting a specific set of images which are designed to identify weak points in algorithm design or performance.

We saw an example of this early in this course with the discovery that CNNs typically identify sheep in landscapes without sheep and misidentify sheep in arms or indoors as dogs or cats.

Some other examples from the literature include:

- Kotseruba *et al.*, 2019, Do Saliency Models Detect Odd-One-Out Targets? New Datasets and Evaluations. BMVC - PDF
- Bruce *et al.*, 2016, A Deeper Look at Saliency: Feature Contrast, Semantics, and Beyond. CVPR - Paper link

Another Example: The Elephant in the Room

Rosenfeld *et al.*, 2018, The Elephant in the Room. arXiv link

Rosenfeld *et al.* tested the affect of inserting an object into an image (such as an elephant) and sliding it across the image to demonstrate interference with the ability of a CNN to detect and localize objects in the image.

A video demonstration is available here.