RECENT TRENDS IN ARTIFICIAL INTELLIGENCE BASED COMPUTER VISION

Hochschule für angewandte Wissenschaft und Kunst
Fakultät für Naturwissenschaften und Technik, Göttingen
Prof. Dr.-Ing. Achim Ibenthal

ibenthal@hawk.de

SoSe 2024

Inhaltsverzeichnis

Nutzungsrichtlinie

Die Inhalte dieser Vorlesung entstammen hauptsdż″chlich den Quellen [1], [2], [3], die als vorlesungsbegleitendes Material empfohlen werden. Weitere Quellen sind im Text kenntlich gemacht. Dardż″ber hinaus gehende Inhalte und Bilder sind © A. Ibenthal.

Die Nutzung dieses Dokumentes dient ausschliedż″lich der Lehre und ist den Studierenden der HAWK, Fachhochschule Hildesheim, Holzminden, Gdż″ttingen fdż″r die Dauer ihrer Immatrikulation gestattet. *Die Weitergabe, Verdż″ffentlichung oder kommerzielle Nutzung – auch nur auszugsweise – ist nicht gestattet.*

Hinweise zu Fehlern, fehlenden oder falschen Referenzen, sowie weitere Kommentare und Anregungen sind stets willkommen und werden zur Verbesserung des Scripts fortlaufend eingearbeitet.

Glossar

Abkürzung	Erläuterung
CG	Computergrafik
CGIW	Computergrafik Interface für Windows®

Konventionen

Texthervorhebungen

Stil	Erläuterung
wichtig!	Hervorhebung wichtiger Begriffe
Datei → Öffnen	Anwendungsmenüs
`a = b^2 + c^2;`	Matlab Quellcode
`fft2`, `semilogy`	Keywords, Funktionsnamen
`% Kommentar`	Kommentar im Quelltext

Textboxen

Rot

- Wichtige Ergebnisse
- Zusammenfassungen

Blau

- Definitionen
- Sätze
- Beweise und Rechnungen

1 Section 1

1.1 Zero-shot learning

zero-shot learning

Definition 1.1: Zero-shot learning.

aims to train a model that can classify objects of unseen classe(target domain) via transfering knowledge obteined from other seen classes(source domain) with the help of semantic information ■

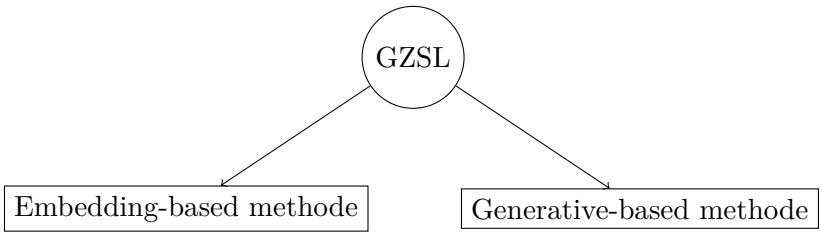
Problem

more challenging scenario with no data for unseen classe,relying solely on descriptions and transfered knowledge from seen classes.

Definition 1.2: Generalized Zero-shot learning.

more realistic scenario with some data for seen classes to help classify unseen classes using description and visual representation ■

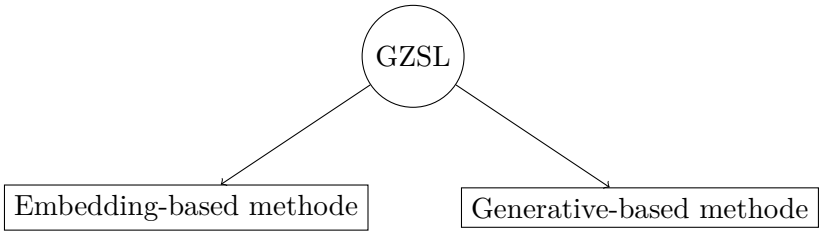
Generalized Zero-Shot Learning (GZSL)



Definition 1.3: Embedding-based methode.

learn an embedding space whether visual-semantic ,semantic-visual, commun/latent or a combination of them (Graph-Based, Autoencoder-Based, Meta-Learning-Based, Compositional Learning and Bidirectional Learning-Based) ■

Generalized Zero-Shot Learning (GZSL)



Definition 1.4: Generative-based methode.

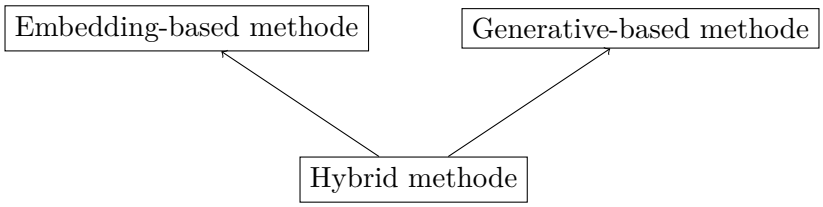
convert GZSL into a conventional supervised learning problem by generating visual features for the unseen classes (Generative Adverserial Network and Variational AutoEncoder (VAE)) ■

Generalized Zero-Shot Learning (GZSL)

	Embedding-based	**Generative-based**
Adventage	• can be effective for simple objects	• can potentially handle complex object better by creating realistic visual representation
Desadvantage	• suffer from the seen classes overfitting problem due to the data imbalance nature of ZSL	• the quality of generated images can be a chalange and impact classification accuracy

Tabelle 1.1: Advantage and Desadvantage of GZSL methodes

Generalized Zero-Shot Learning (GZSL)



- The Constractive Embedding for generalized zero-Shot learning paper proposes a hybrid GZSL framwork that comnbines the two models

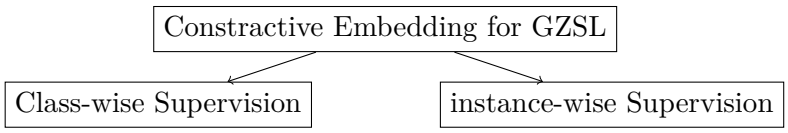
Definition 1.5: Constractive Embedding.

is integrated into hybrid GZSL with the goal to construct embeddings that facilitate the recognition of unseen classes by leveraging the relationships and similarities between seen and unseen classes ■

Generalized Zero-Shot Learning (GZSL)

Definition 1.6: Constractive Embedding.

is integrated into hybrid GZSL with the goal to construct embeddings that facilitate the recognition of unseen classes by leveraging the relationships and similarities between seen and unseen classes ■



Definition 1.7: Class-wise and instance-wise supervision .

Class-wise Supervision is to classify the membership and the instance-wise Supervision is to identify the similarities or the differenz between individual image . ■

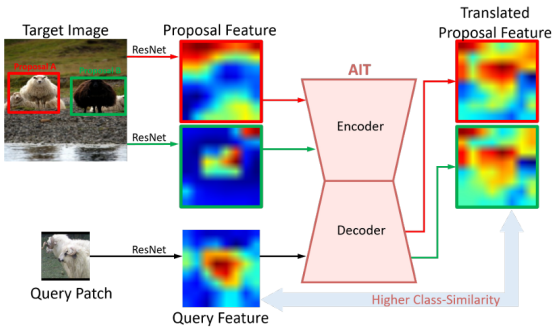
2 Section 2

2.1 One-Shot learning

One-shot learning

Definition 2.1: One-shot learning .

aims to identifying within a target image all object instance of the same class,implied by a query image patch with the problem that its label and its respective exemple are not available in the training data . ■



Definition 2.2: Adaptive Image Transformer .

AIT module aims to compare the features of regions or abjects in the Target Image based on the language translation process: ■

One-shot learning

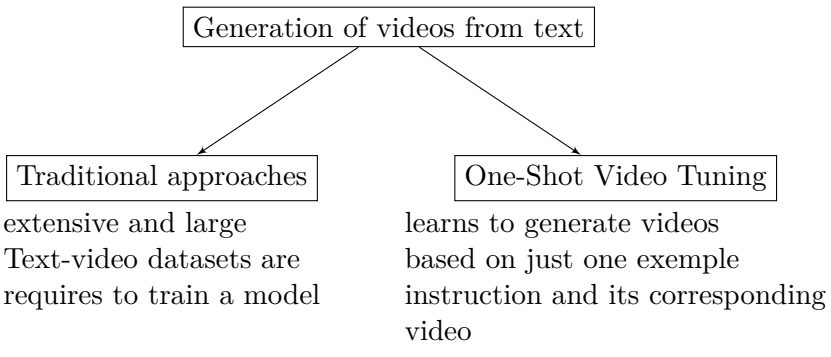
- AIT can adaptively represent each region proposal so that the similarity with the query can be evaluated ,that module uses an attention based encoder-decoder architecture to simultaneausly explore intra-coder and inter-coder
- encoder converts the inpute data (e.g. features of objects proposals) into internal representation.
- decoder generates the output from this internal representation(e.g. class similarity between proposals and the query image.)
- Intra-coder attention refers to the relationships within a single coder, for exemple the relationship and the importance between different parts of a single image proposal to generate a better internal representation.
- Inter-coder attention examines the relationshipes between different coders. the model learns how each proposal is related the query image,which help to better judge class similarity

One-shot learning

- sparse attention mecanism help the model focus on the most important part of the data.

Evaluation Higher mAP (Mean Average Precision) means more accurately detecting and classifying abjects from all classes .

One-shot learning in Text-to-video generation



One-shot learning in Text-to-video generation

One-shot video tuning

```
" a man is running on the beach" and a single video that
shows exactly this scene .
```

1. pretraining phase : learns to generate images from text description
2. learn how to translate the specific text description into motion pictures, das model learns that the beach describe a specific environment and a man has a certain shape
3. generating similar video : generation of videos based on different text description with different scenario , event though had learnt just from one exemple

Application

1. object editing .
2. Background changing .
3. generating similar video : generation of videos based on different text description with style transfer for exemple from real-worls into comic style.

3 Section 3

3.1 Few-Shot learning

Few-shot learning

Definition 3.1: Few-shot learning .

aims to learn information about Data from very limited number of training data when the collecting a large dataset is impossible . ■

Problem

more challenging for few-shot learning in traditional neural network is the low effectiveness with very limited labled data (e.g. overfitting).

Definition 3.2: Overfitting .

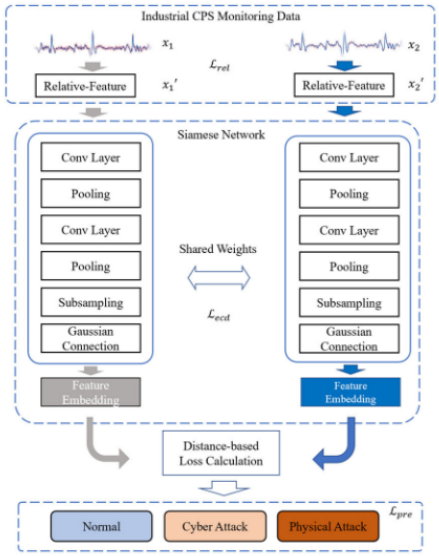
A model overfits the training data when it describes features that arise from noise or variance in the data, rather than the underlying distribution from which the data were drawn. Overfitting usually leads to loss of accuracy on out-of-sample data. ■

Few-shot learning

Definition 3.3: A few-shot learning model based on Siamese Convolutional Neural Network (CNN) .

A Siamese CNN consists of two identical subnetworks that work in parallel and share the same weights. This network compares two inputs and learns whether they belong to the same class or not. By comparing sample pairs, the model can learn more robust and generalizable features, which reduces overfitting. ■

Few-shot learning



- pooling layer helps in reducing high dimensional convolutional features .
- Conv layal to identify from the imput data the importent features using edges and textures detection .
- two combinations of convolution layer and pooling layer are introduced to extract feature embeddings

Few-shot learning

- the distance between these two feature embeddings will be calculated to identify whether these two input samples belong to the same class.

Few-shot learning

Definition 3.4: Few-Shot Hyperspectral Image Classification .

Hyperspectral Image Classification : The objects are classified by its HyperSpectrum .

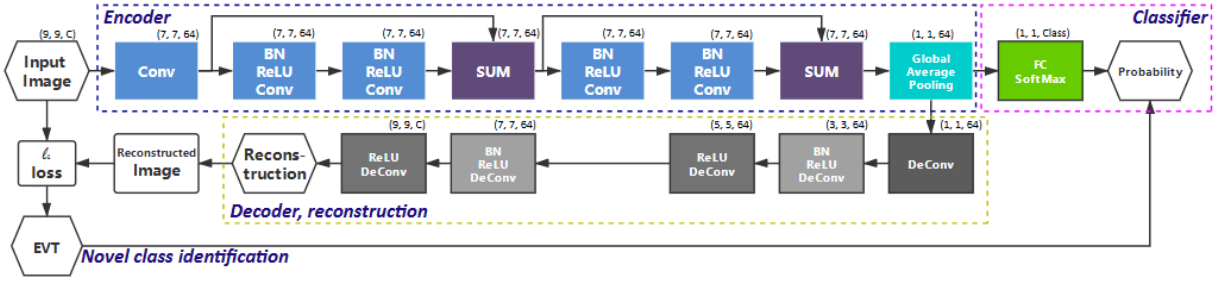
Problem

Current hyperspectral image classification assumes that a predefined classification system is closed and complete, and there are no unknown or novel classes in the unseen data. When a new class appears, it will be interpreted as an error ,what is not the case .

Definition 3.5: MDL4OW(multitask deep learning).

method that simultaneously conducts classification and reconstruction in the open world .The reconstructed data are compared with the original data; those failing to be reconstructed are considered unknown classes .

Few-shot learning



- the encoder/features extractor get from the inpute image the proposal features using Pooling and convolution layal .
- after extracting the latent features the function softmax serves as the classifier and outputes the probability to the known classes .
- the reconstruction task uses the deconvolutional layers to increase the spatial dimonsion of the latent features gradually, the output should be similar to the input data .

ďž˝bung 3.1: ...

...

a) ...

b) ...

c) ...

ďž˝bung 3.2: ...

...

a) ...

b) ...

c) ...

4 Section 2

4.1 Subsection 1 of Section 1

Frame 1 of Subsection 1 of Section 2

...

4.2 Übungen zu Abschnitt 4

dž″bung 4.1: ...

...

a) ...

b) ...

c) ...

5 Section 3

5.1 Subsection 1 of Section 3

Frame 1 of Subsection 1 of Section 3

...

5.2 Übungen zu Abschnitt 5

dž″bung 5.1: ...

...

a) ...

b) ...

c) ...

6 Layout Options

6.1 General Hints

- The following files have been created to enhance the flexibility of the `beamer` class in terms of layout and to provide commands which facilitate its application especially for *lectures*:
  - *`settings_common.tex`*: Things which are common to both slide and script. Loads packages, sets layout and theme colours, adds commands and environments, switches PDF realtime clock for slides on/off, sets author and facility information.
  - *`settings_slides.tex`*: slide-specific layout: defines 35 additional color themes, new logo layout, colour control for theme elements.
  - *`settings_slides.tex`*: script-specific layout such as page layout and boxes.
  - *`mybeamer.sty`*: commands shortcuts for standard text colors, beamerclass control, figures, boxes and math.
- To adapt the *overall layout* use Section 2 of `\format\settings_common.tex`.
- To adapt *lecture title*, *personal data* and *legal notice* use *Section 3* of `\format\settings_common.tex`.
- Commands for *blocks* and *boxes* are explained in *this Section*.
- `\te{}` provides *text emphasis*, `\ce{}` character emphasis (e.g. for glossary) and `\me{}` the marking of menue commands. It is recommended to use those because the layout is optimized for slides and script individually. Defaults are defined in Section 10 of `\format\settings_common.tex`. A `\key{}` command is available to add sidenotes to the script.
- `beamer` increments counters in `\section*`, `\subsection*`, `\subsubsection*`. Therefore use new commands `\nosection`, `\nosubsection`, `\nosubsubsection` instead (not starred).
- *`mybeamer.sty`* defines many additional commands, which are more or less selfexplanatory when reading `mybeamer.sty`. Examples:
  - Color definitions.
  - commands which operate only in one of the two modes slide and script such as `\slideonly{any text}` and `\scriptonly{any text}`.
  - e.g. `\slup{#}` and `\scrdown{#}` are used for v- and hspaces. Arguments `#` are passed without unit and interpreted as '`ex`'.
  - `\switch{#1}{#2}` separates arguments used in slide mode (`#1`) and script mode (`#2`). This comes in quite handy e.g. for figure resizing.
  - Often a `\vspace` is overruled by the TEX auto layout and more or less ignored. Here the commands `\slidebar{#}` and `\scriptbar{#}` are providing the brute force method to enforce spacing.

– `\fig` facilitates the insertion of figures:

```
\fig{captiontext}
    {fig:<label>}
    {\switch{0.9}{0.7}} % sizes for slide/script
    {figurename}        % some pdf, jpg or png
    \scriptbar{2}       % enforce space in script
```

– There are many more, check it out!

6.2 Blocks and Boxes

`normalblock` and `normalbox` Environment

normalblock Title

```
\begin{normalblock}{normalblock Title}
   ...
\end{normalblock}
```

- normalblock item

1. normalblock enumerate

```
\begin{normalbox}
   ...
\end{normalbox}
```

- normalbox item

1. normalbox enumerate

`exampleblock` and `examplebox` Environment

exampleblock Title

```
\begin{exampleblock}{exampleblock Title}
   ...
\end{normalblock}
```

- examplelblock item

1. exampleblock enumerate

```
\begin{examplebox}
   ...
\end{examplebox}
```

- examplebox item

1. examplebox enumerate

`alertblock` and `alertbox` Environment

alertblock Title

```
\begin{alertblock}{alertblock Title}
   ...
\end{alertblock}
```

- alertblock item

1. alertblock enumerate

```
\begin{alertbox}
   ...
\end{alertbox}
```

- alertbox item

1. alertbox enumerate

`block` and `bodybox` Environment

The standard `block` environment defined by `beamer` uses theme colors for title, title text, canvas, body text and bullets:

Theme

```
\begin{block}{block Title}
   ...
\end{block}
```

- block item

1. block enumerate

We have added a `bodybox` environment. It's effect is similar to a `beamercolorbox` with `block body` color (but vertically more 'tight'):

```
\begin{bodybox}
   ...
\end{bodybox}
```

- bodybox item

1. body enumerate

Shortcuts

It turns out that in practice typing and error rate can be reduced by using *shortcuts* for the block and box environments. In `article` mode the `\nblock`, `\nbox`, `\ablock` and `\nbox` commands can be colored too, using the `\def\scriptboxlayout{boxed}` option in `settings_common.tex`, Section 2.

```
\nblock{Title}{Body}
\nbox{Body}
\eblock{Title}{Body}
\ebox{Body}
\ablock{Title}{Body}
\abox{Body}
\tblock{Title}{Body}
\tbox{Body}
```

Examples:

```
\nblock
   {Problem}
   {Does dark matter smell?}
```

Problem

Does dark matter smell?

```
\abox{\centering%
      This is important!}
```

This is important!

`beamercolorbox` Environment

The `beamercolorbox` environment provides boxes which' canvas is derived from some beamer color. *Text and bullet colors* are derived from the beamer theme colors (*!*), however. Better use

- `\nbox` for blue boxes (n = normal)

- `\ebox` for green boxes, also used for examples

- `\abox` for alert boxes

- `\tbox` for theme color box,

giving you matching bullet colors in presentetion (= beamer) mode and black bullets in article mode anyway.

6.3 New Environments

Definitions `\ndef`

The beamer `definition` and `Definition` environments are using template colors for boxes, too. In `settings_common.tex` a *new environment for definitions* is created which uses blue boxes. This is recommended, since otherwise the audience has to adapt to different box colors in case you are just changing the Theme color.

```
\ndef{def:<label>}{Analogkäse}{
    Das willst Du nicht wirklich wissen!
\filledend}
```

Definition 6.1: Analogkäse.

Das willst Du nicht wirklich wissen! ■

- We don't provide numbers in presentation (= beamer) mode because
  - they don't mean much to the audience
  - and because they are causing problems when using the beamer overlay specifications such as `\uncover<>`.
- Numbering in script is local to Sections.
- Note that we had to come up with something else than `\definition` or `\Definition` since those were already taken. Thus: `\ndef`
- The 'n' indicates that normal block color is being used.

Satz `\nsatz` and Theorem `\ntheorem`

`\nsatz` (Satz) and `\ntheorem` (Theorem) essentially are working in the same way as `\ndef`, except they are producing blue boxes in the script if script boxes are enabled in Section 2 of `settings_common.tex`.

Theorem 6.1: Berti's Sicht.

Die Realität sieht anders aus als die Wirklichkeit.

Satz 6.1: Fourier-Transform.

$$S(\omega) = \int_{-\infty}^{\infty} s(t)e^{-j\omega t}dt, \tag{6.1}$$

$$s(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S(\omega)e^{j\omega t}d\omega. \tag{6.2}$$

Korollar `\ncor` and Lemma `\ntheorem`

`\ncor` (Korollar)/`\ncor` (Corollary) and `\nlem` (Lemma) do not produce boxes in script mode.

Korollar 6.1: Osterhasenpädagogik.

Der Lehrer versteckt das Wissen - die Schüler sollen's suchen. □

Corollary 6.2: Maturity.

Growing old is mandatory, growing up is optional. □

Lemma 6.1: Jordan's Lemma.

In der Elektrotechnik gibt es Probleme, bei denen komplexe Zahlen so nützlich sind, wie negative Zahlen für Probleme mit Geld. □

The white boxes "□" in script mode are generated using `\openend`.

Exercises: `\Uebung` – `\Antwort` environment pair

For the application of the `\Uebung` – `\Antwort` *environment pair* for exercises and answers see the LaTeX source code of this template. You can change the environment title and language of headings in the environment definitions in Section 7 of `settings_common.tex`. Numbering in script is local to Sections.

6.4 Sourcecode Listings

Source code listings are inserted by using the *listings* package. This template is configured for Matlab M-file language. Use Section 11 of settings_common.tex if you wish to change the default.

Example:

```
\begin{lstlisting}[title=\cblueb{averagingdemo.m},
                   label=list:average,
                   firstnumber=1,
                   backgroundcolor=\color{ultralightgray}]
% Demo Planarer Tiefpass

%% Bild lesen
b  = imread('zoneplate.tif');

%% Filterung
b5  = imfilter(b, fspecial('average', 5),  'replicate');
b9  = imfilter(b, fspecial('average', 9),  'replicate');
b13 = imfilter(b, fspecial('average', 13), 'replicate');

%% Display
subplot(2,2,1); imshow(b,  []);  title('Original');
subplot(2,2,2); imshow(b5, []);  title('5 x 5 Filter');
subplot(2,2,3); imshow(b9, []);  title('9 x 9 Filter');
subplot(2,2,4); imshow(b13, []); title('13 x 13 Filter');
\end{lstlisting}
```

averagingdemo.m

```
1  % Demo Planarer Tiefpass
2
3  %% Bild lesen
4  b  = imread('zoneplate.tif');
5
6  %% Filterung
7  b5  = imfilter(b, fspecial('average', 5),  'replicate');
8  b9  = imfilter(b, fspecial('average', 9),  'replicate');
9  b13 = imfilter(b, fspecial('average', 13), 'replicate');
10
11 %% Display
12 subplot(2,2,1); imshow(b,  []);  title('Original');
13 subplot(2,2,2); imshow(b5, []);  title('5 x 5 Filter');
14 subplot(2,2,3); imshow(b9, []);  title('9 x 9 Filter');
15 subplot(2,2,4); imshow(b13, []); title('13 x 13 Filter');
```

7 Lösungen zu den Übungen

Lďž″sung zu dž″bung 3.1

Solution of first exercise in Section 1

Lďž″sung zu dž″bung 3.2

Solution of second exercise in Section 1

Lďž″sung zu dž″bung 4.1

Solution of exercise in Section 2

Lďž″sung zu dž″bung 5.1

Solution of exercise in Section 3

Literatur

[1] W.-P. Buchwald, “Audio-Video-Systeme 2”, Vorlesungsskript FH Braunschweig/Wolfenbüttel, 2005.

[2] B. Wendland, *Fernsehtechnik - Band 1: Grundlagen*, Dr. Alfred Hüthig Verlag GmbH Heidelberg, 1988.

[3] B. Wendland, *Fernsehtechnik - Band 2: Systeme und Komponenten zur Farbbildübertragung*, Dr. Alfred Hüthig Verlag GmbH Heidelberg, 1988.