Optical Character Recognition (OCR)

- CV

  Tesseract 3 (A*)
  Binarizuje, naporcuje obrazok na kúsky a snaží sa z nich poskladat symboly

- CNN+RNN

  Pre každý pixel povie pravdepodobnosť že je súčasťou textu, z oblastí z vysokou pravdepodobnosťou spraví obdĺžniky a tieto rozpoznáva rekurentnou neurónovou sieťou

- Holistic

  Zákoduje obraz na príznakovú mapu a z nej generuje texty (transformer typu encoder-decoder)

Vstup

technical details are too complex to cover in the book itself.

In teaching our courses, we have found it useful for the students to attempt a number of small implementation projects, which often build on one another, in order to get them used to working with real-world images and the challenges that these present. The students are then asked to choose an individual topic for each of their small-group, final projects. (Sometimes these projects even turn into conference papers!) The exercises at the end of each chapter contain numerous suggestions for smaller mid-term projects, as well as more open-ended problems whose solutions are still active research topics. Wherever possible, I encourage students to try their algorithms on their own personal photographs, since this better motivates them, often leads to creative variants on the problems, and better acquaints them with the variety and complexity of real-world imagery.

In formulating and solving computer vision problems, I have often found it useful to draw inspiration from three high-level approaches:

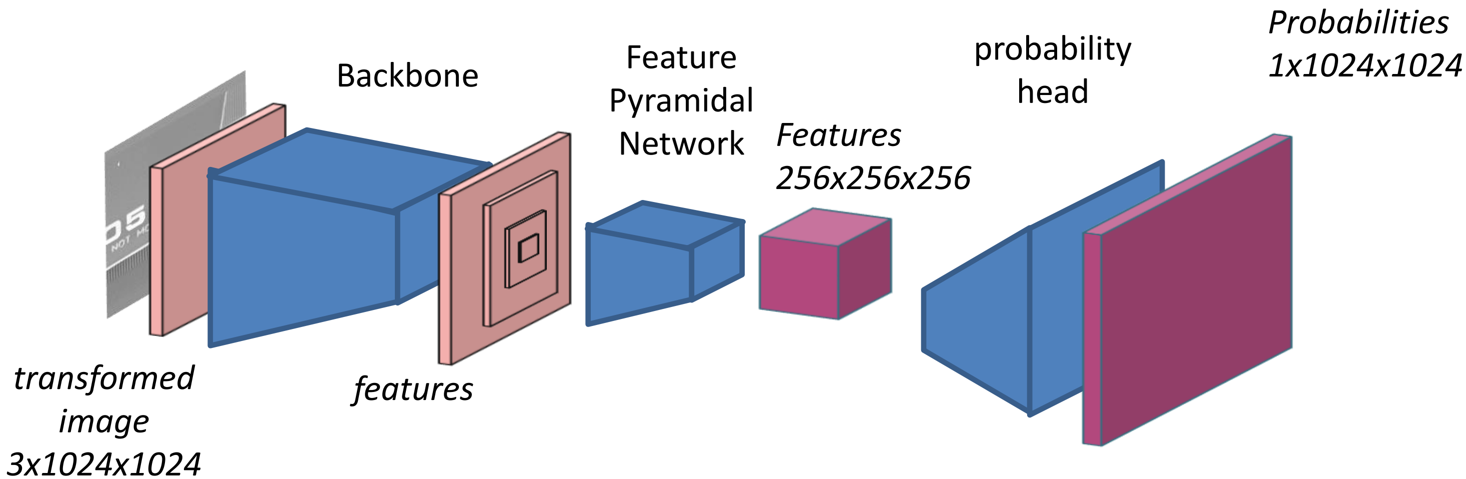
CNN+RNN

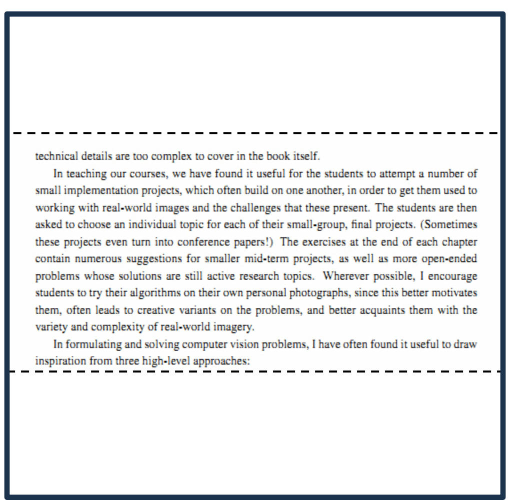
OCR tohto druhu sa skladá z dvoch hlbokých modelov:

- Detektor textu (konvolučná sieť)
- Rozpoznávač textu (dvojsmerná rekurentná sieť)

Výstup prvého je so vstupom druhého prepojený algoritmami klasického počítačového videnia

Detektor textu

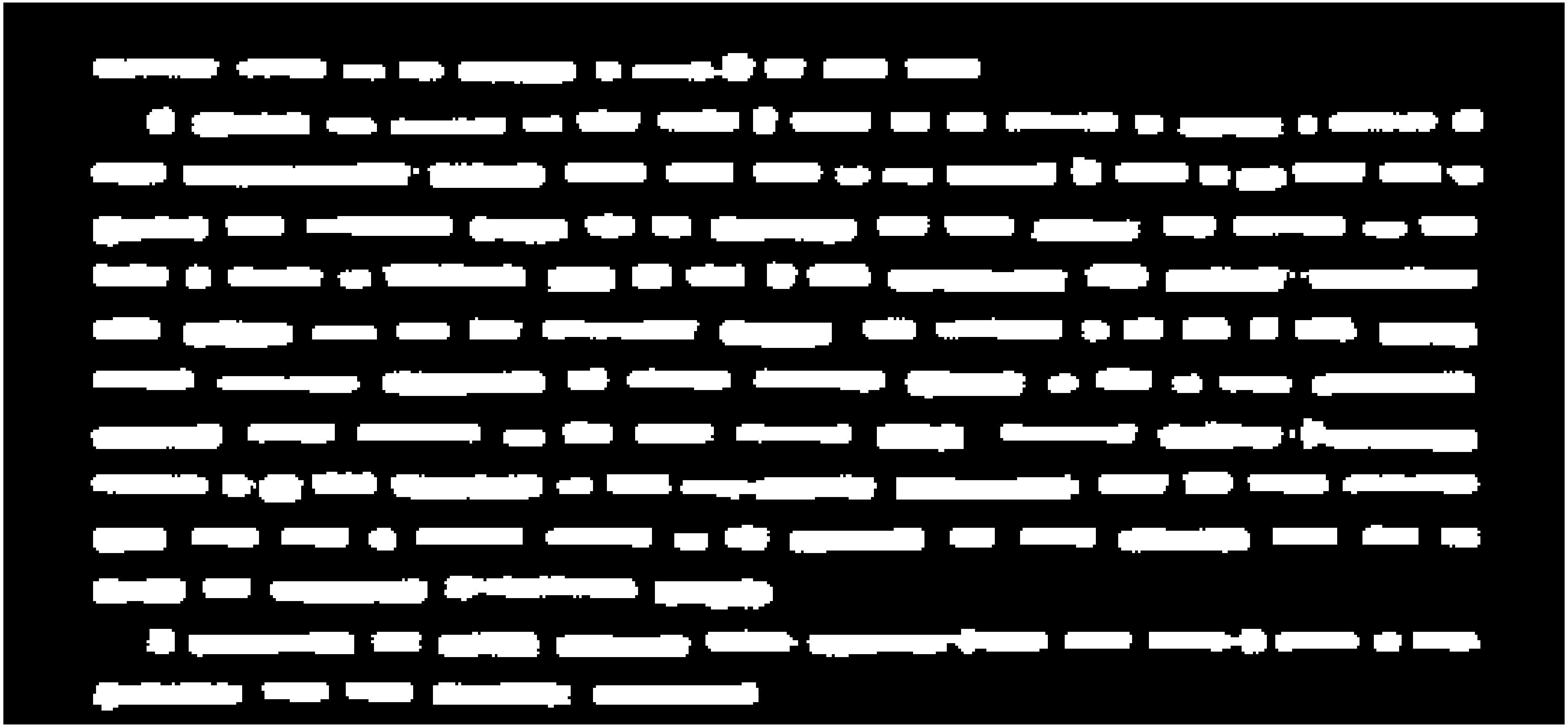



technical details are too complex to cover in the book itself.

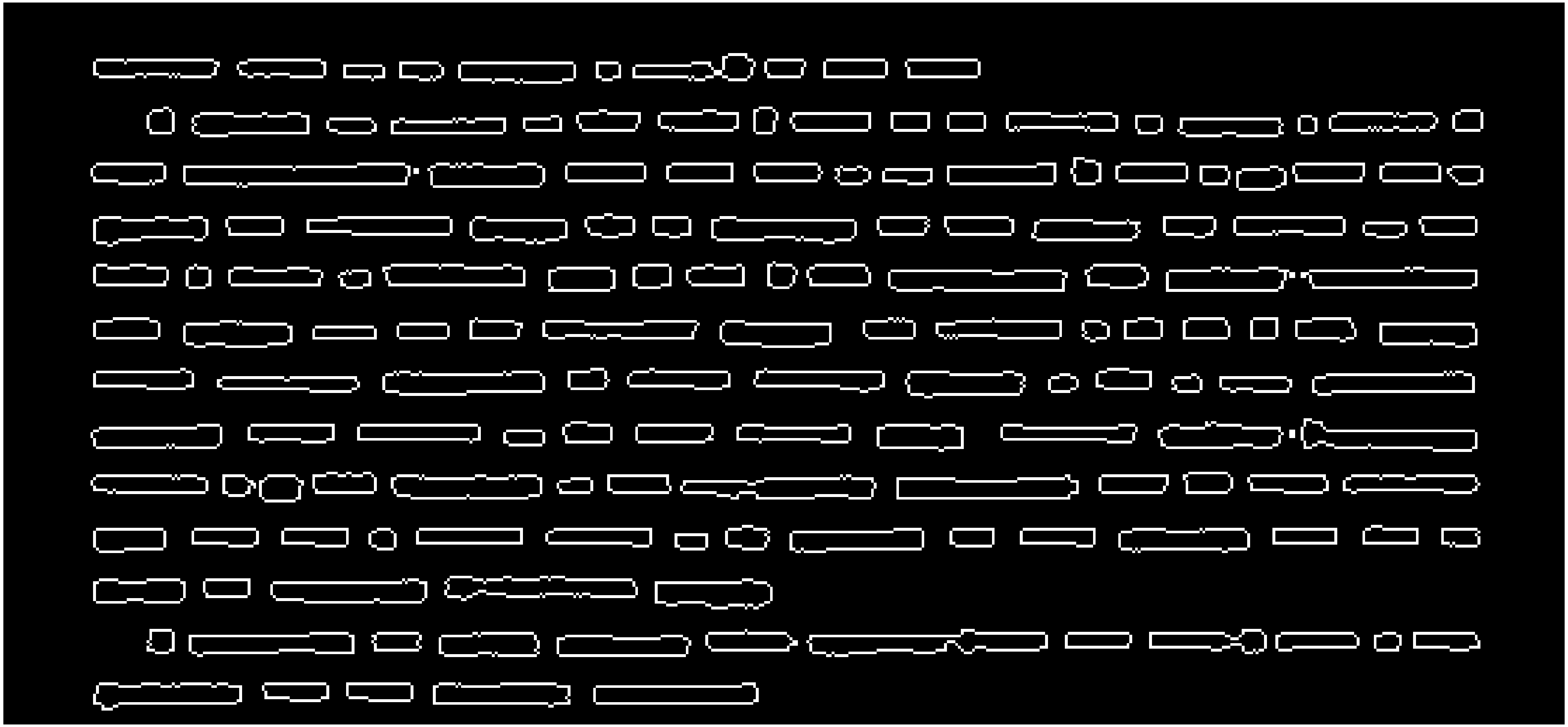
In teaching our courses, we have found it useful for the students to attempt a number of small implementation projects, which often build on one another, in order to get them used to working with real-world images and the challenges that these present. The students are then asked to choose an individual topic for each of their small-group, final projects. (Sometimes these projects even turn into conference papers!) The exercises at the end of each chapter contain numerous suggestions for smaller mid-term projects, as well as more open-ended problems whose solutions are still active research topics. Wherever possible, I encourage students to try their algorithms on their own personal photographs, since this better motivates them, often leads to creative variants on the problems, and better acquaints them with the variety and complexity of real-world imagery.

In formulating and solving computer vision problems, I have often found it useful to draw inspiration from three high-level approaches:

Výstup detektora

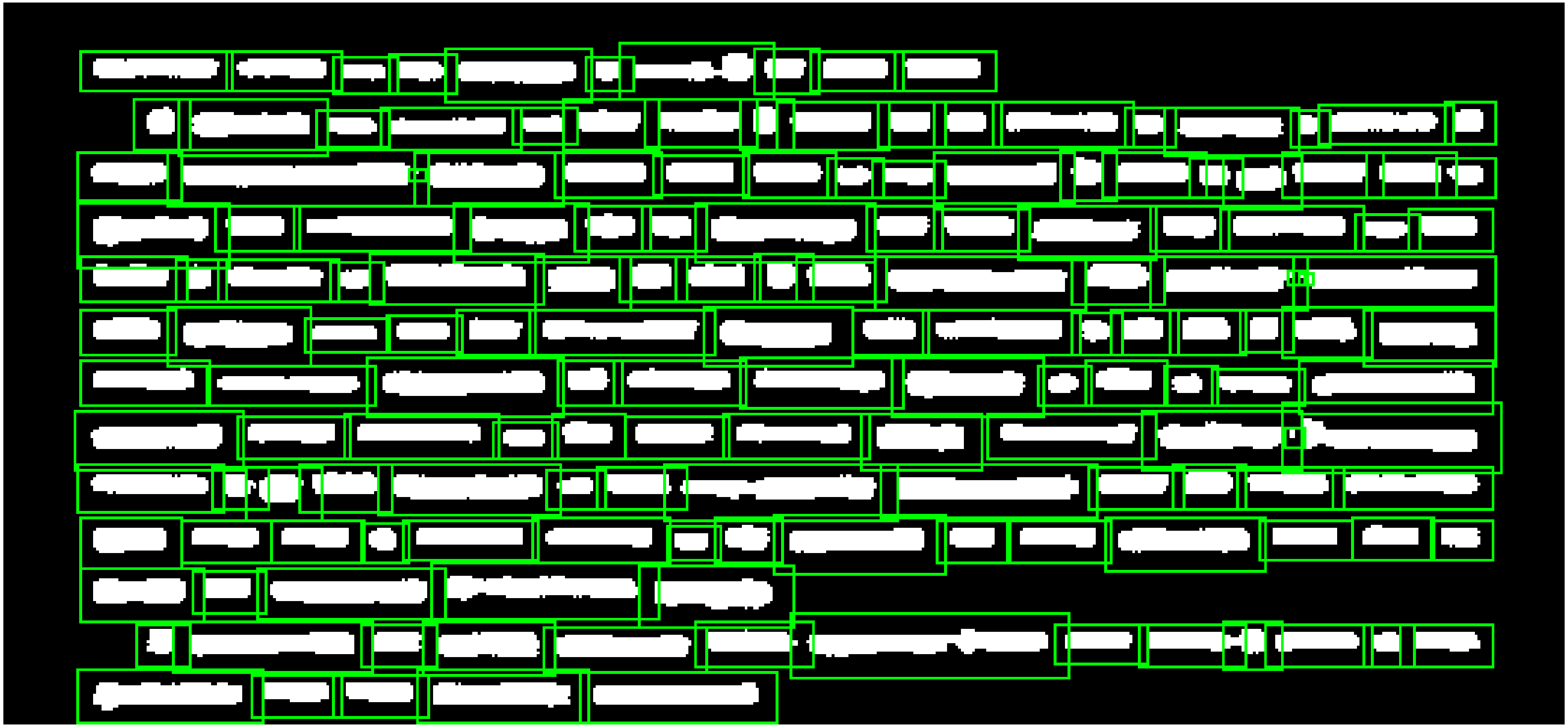


Kontúry



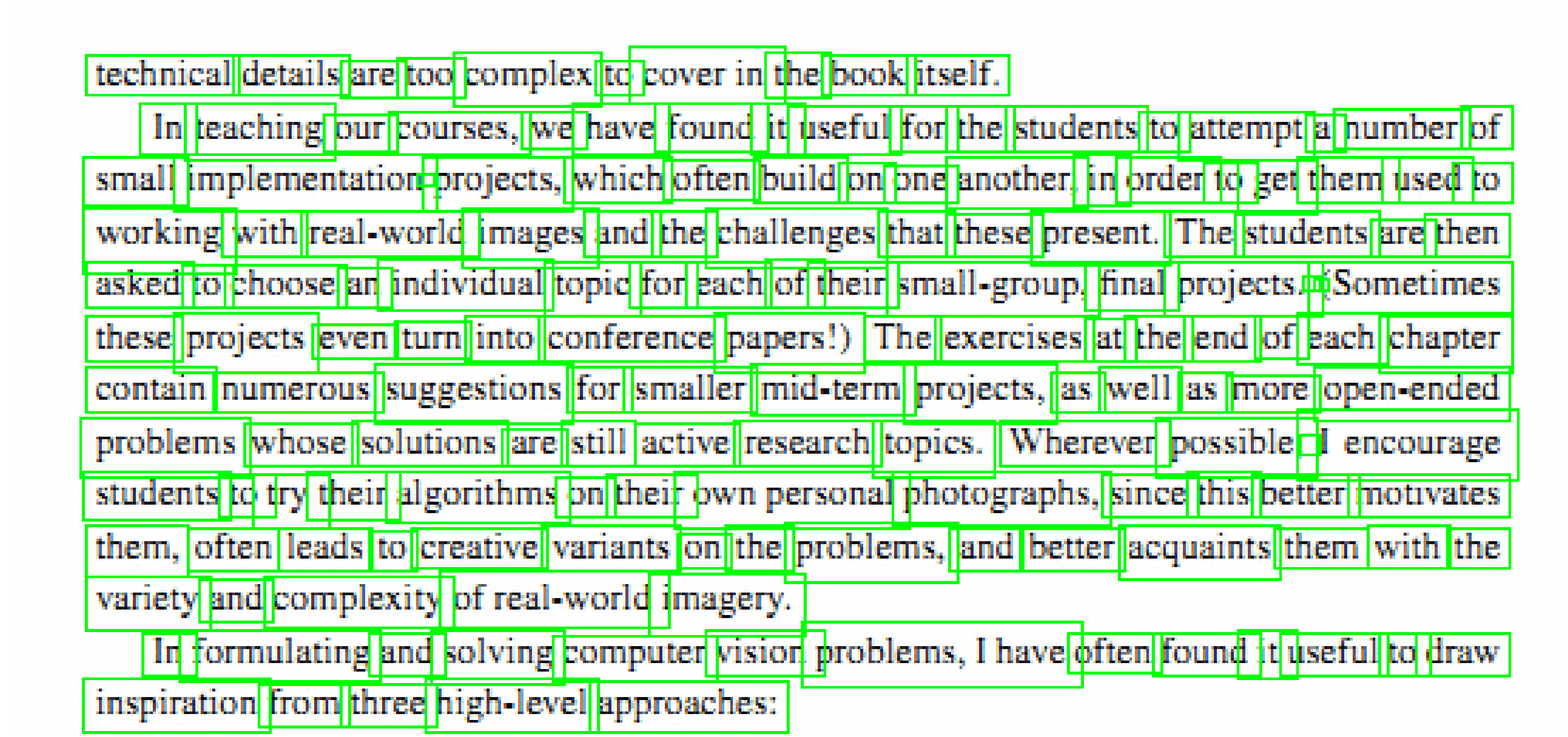
Kontúry môžeme aproximovať štyrmi alebo viacerými bodmi

Obdĺžniky alebo Polygóny



Pritom oblasti zodpovedajúco zväčšíme, aby sme zachytili celý text

Výsledok spracovania detekcie

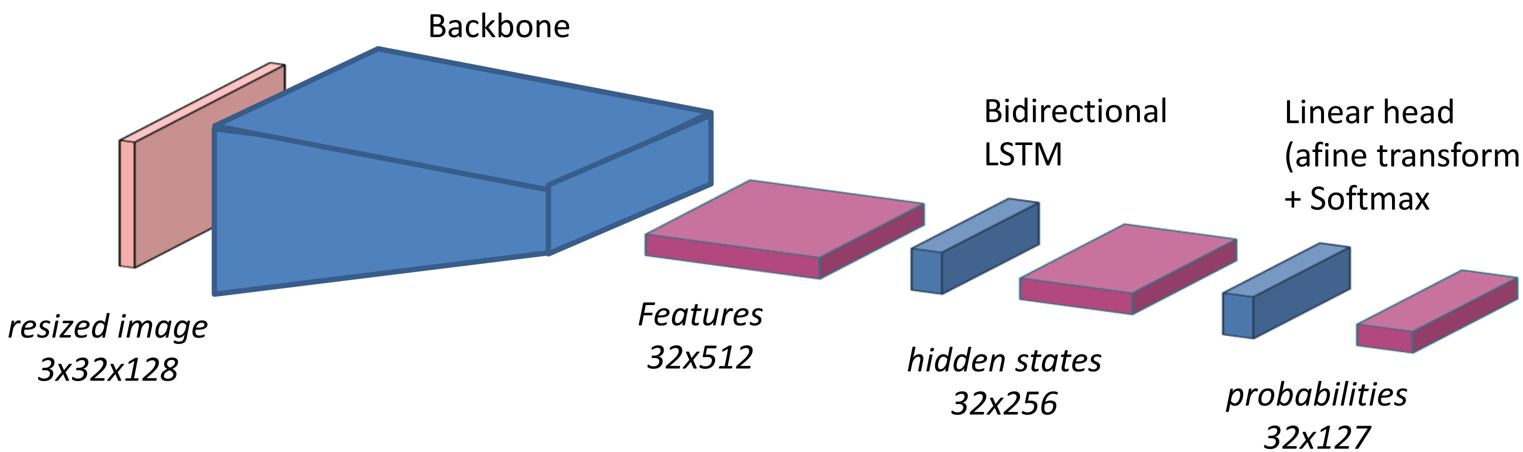

technical details are too complex to cover in the book itself.

In teaching our courses, we have found it useful for the students to attempt a number of small implementation projects, which often build on one another, in order to get them used to working with real-world images and the challenges that these present. The students are then asked to choose an individual topic for each of their small-group, final projects. (Sometimes these projects even turn into conference papers!) The exercises at the end of each chapter contain numerous suggestions for smaller mid-term projects, as well as more open-ended problems whose solutions are still active research topics. Wherever possible, I encourage students to try their algorithms on their own personal photographs, since this better motivates them, often leads to creative variants on the problems, and better acquaints them with the variety and complexity of real-world imagery.

In formulating and solving computer vision problems, I have often found it useful to draw inspiration from three high-level approaches:

Každý obĺžnik či polygón vyjmene z textu a budeme rozpoznávať

Rozpoznávač



vocab (126 characters):

0123456789abcdefghijklmnopqrstuvwxyzABCD
EFGHIJKLMNOPQRSTUVWXYZ!"#$%&'()*+,-
./:;<=>?@[\]^_`{|}~°£€¥¢฿
àâéèêëîïôùûüçÀÂÉÈÊËÎÏÔÙÛÜÇ

Rozpoznávanie

- Vyrežeme kúsok textu

- Upravíme jeho veľkosť na 32x128

- Tento obraz premeníme na mapu príznakov 32x512
- Tú dekódujeme cez LSTM (typ RNN) na skryté stavy 32x256
- Lineárnou hlavou ich premeníme na 32xN logitov, kde N je počet znakov + 1 (BLANK), aplikujeme softmax
- Cez CTC z pravdepodobností vyberieme symboly

Problém s rýchlosťou

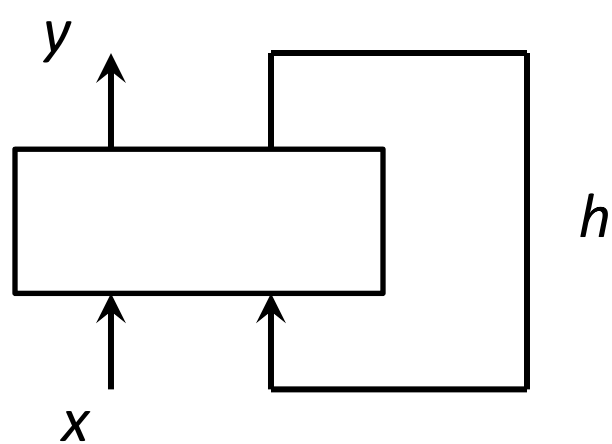
- Dvojsmerná (bidirectional) RNN je schopná pracovať na texte rôznej dĺžky (je naťahovacia)
- Ale nevie pracovať na textoch rôznej dĺžky naraz v jednej dávke
- Preto DocTR radšej naseká texty tak, aby mali menej ako 32 symbolov a používame rovnako veľký vstup 3x32x128
- (Ale Pytorch vie pustiť rôzne dĺžky naraz, akurát vyžaduje aby bola dávka utriedená podľa dĺžky zostupne takže treba urobiť správnu permutáciu)

Výstup

technical details are too complex to cover in the book itselt.
In teaching our courses, we have found it usetul for the students to attempt a number of small implementation projects, which often build on one another, in order to get them used to working with real-world mages and the challenges that these present. The students are then asked to choose an individual topic for each of their small-group, final projects. (Sometimes these projects eveu turn into conference papers!) The exercises at the end of each chapter contain numerous suggestions for smaller mid-term projects, as well as more open-ended problems whose solutions are still active research topics. Wherever possible, I encourage students to try their algorithms on their own personal photographs, since this better motivates them, often leads to creative vanants on the problems, and better acquaints them with the variety and complexity of real-world imagery.
In formulating and solving computer vision problems, I have often found it useful to draw inspiration from three high-level approaches:

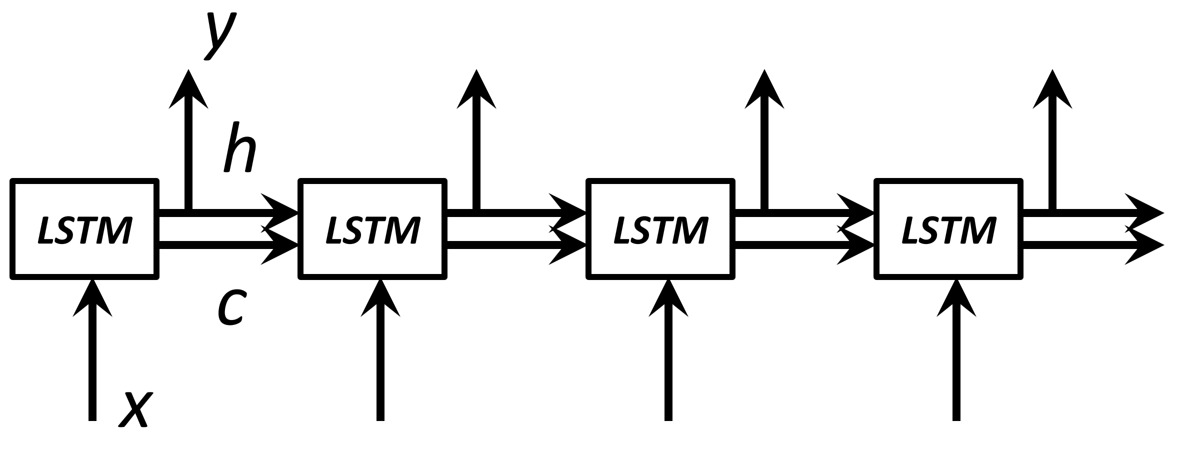
Rekurentná neurónová sieť

- Sieť inšpirovaná spracúvaním postupnosti dát v čase
- V každom okamihu dostáva nový vstup, dáva nový výstup a skrytý stav sa zapamätá a bude pridaný na vstup v ďalšom okamihu



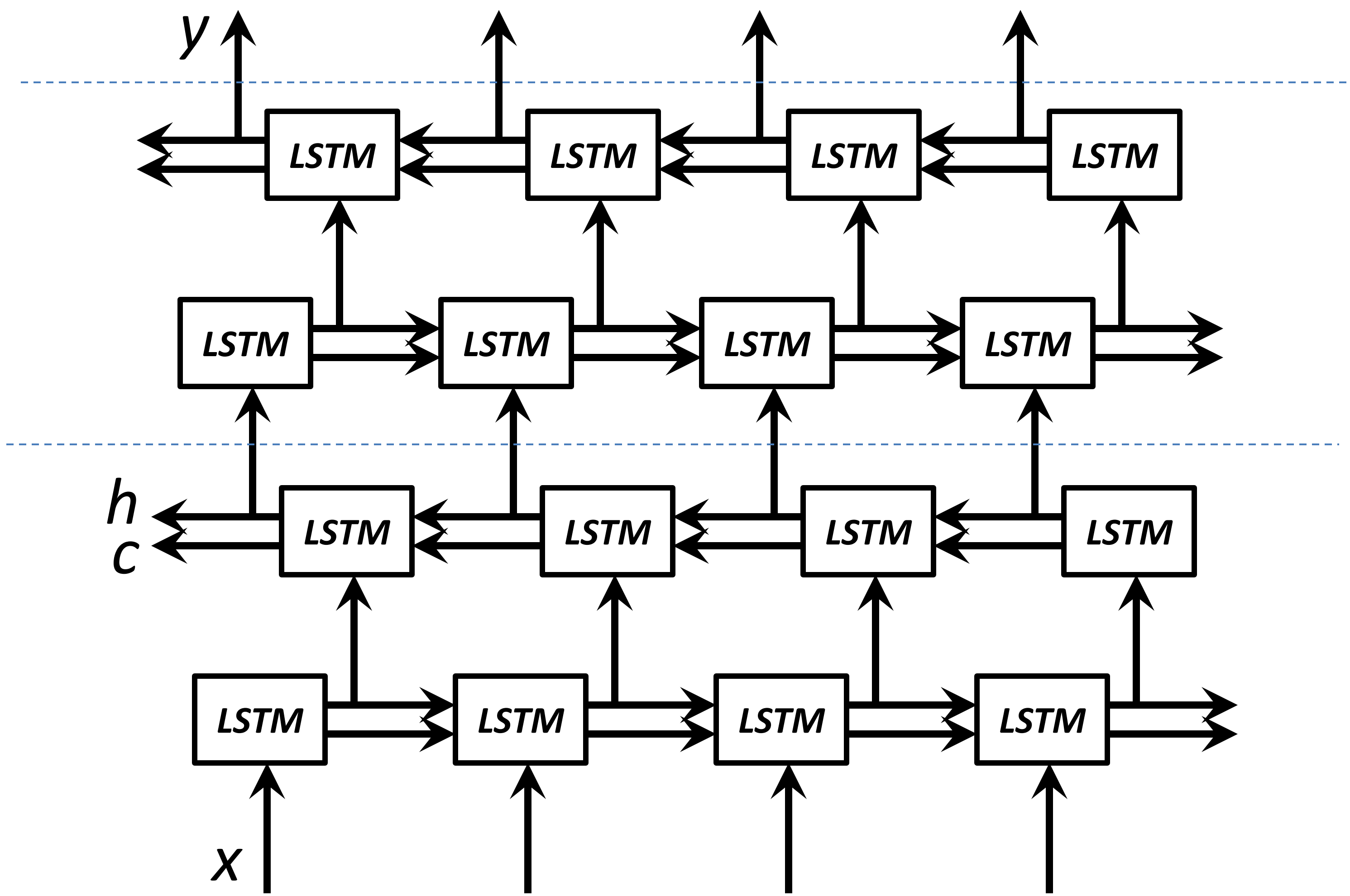
RNN

- Reálne sa nepúšťa na dáta postupne v čase, ale na konci naraz



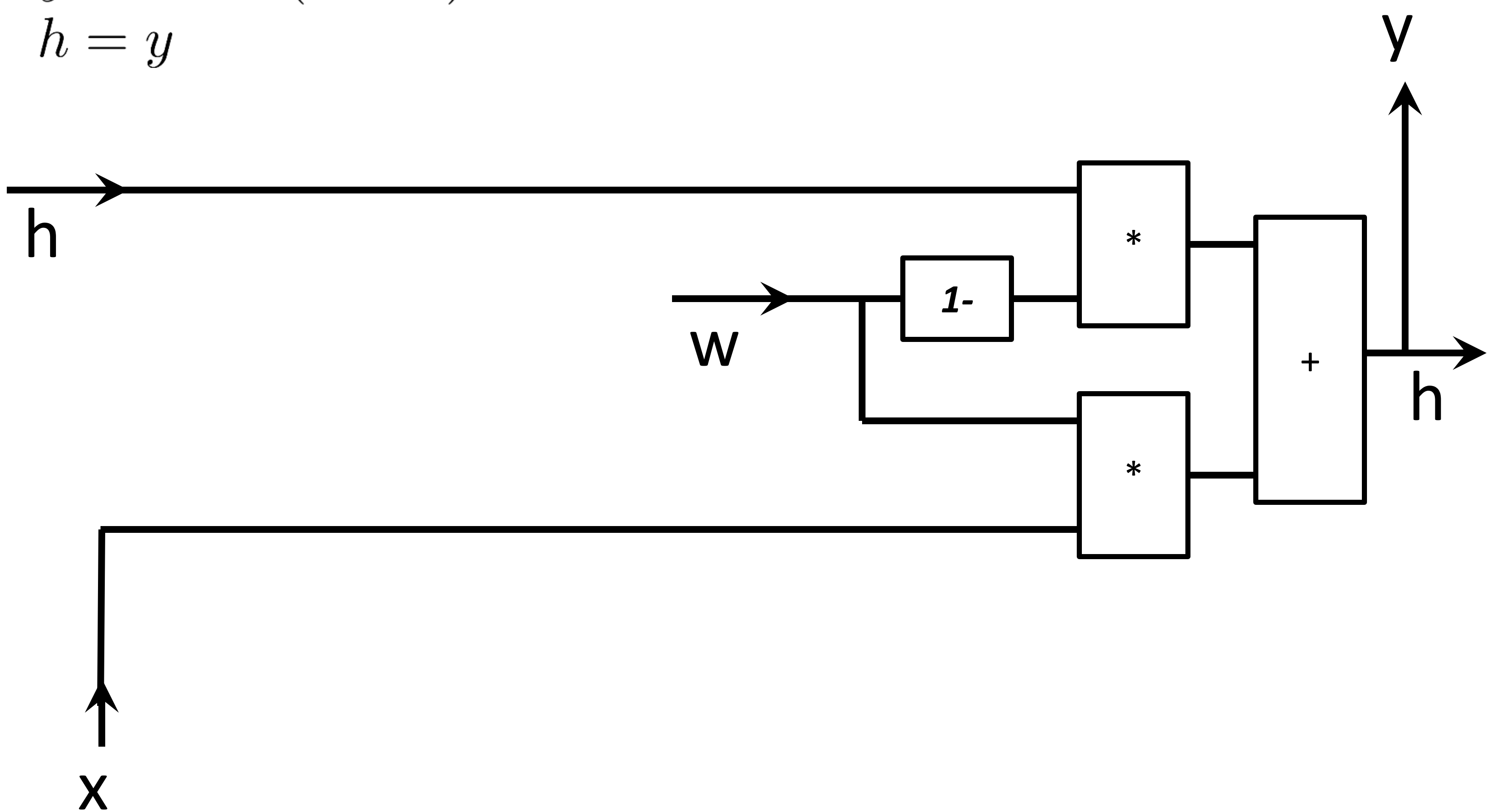
- počet LSTM modulov je daný dĺžkou vstupu

Bidirectional RNN

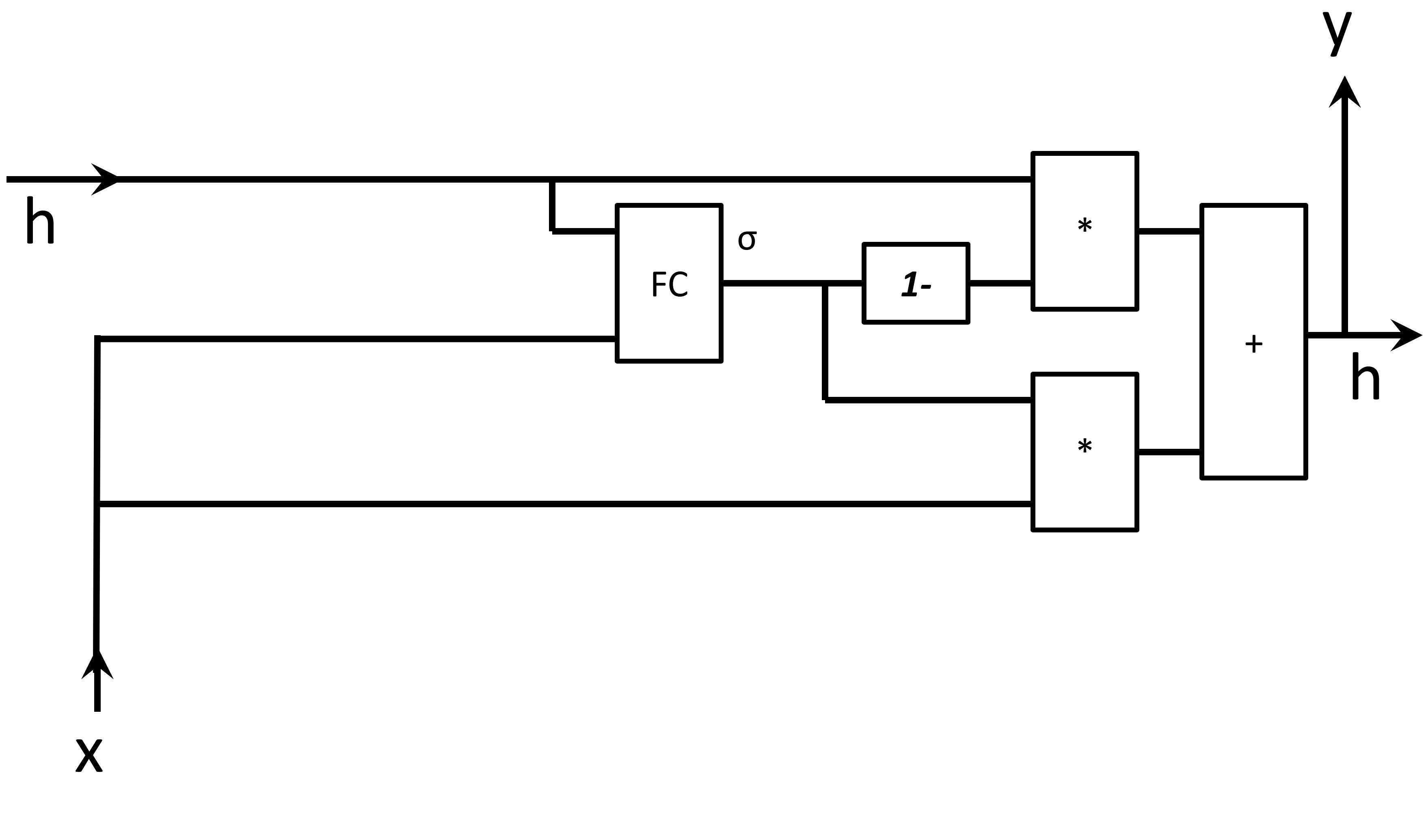


```
x: -1,-1,-1,-1,-1,-1,1,1,1,1,-1,1,1,1,1,1,1,-1,1,1,1,1,1…
y: 0,-0.4,-0.7,-0.9,-1,-0.9,-0.7,-0.4,0.1,0.5,0.9,1,1,1,1,…
```

$$y = wx + (1-w)h$$
$$h = y$$

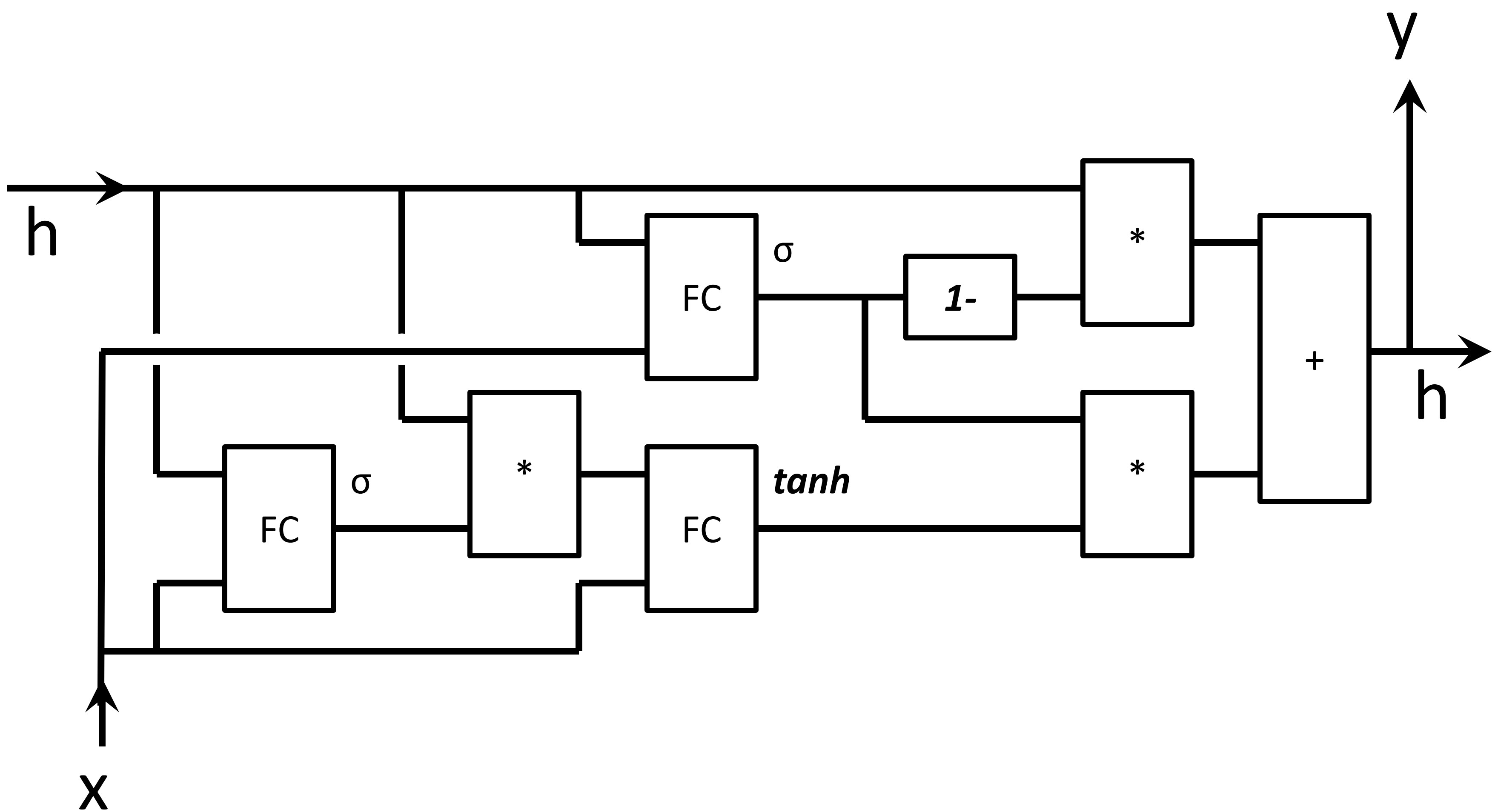


dimension of x, y and h can be > 1

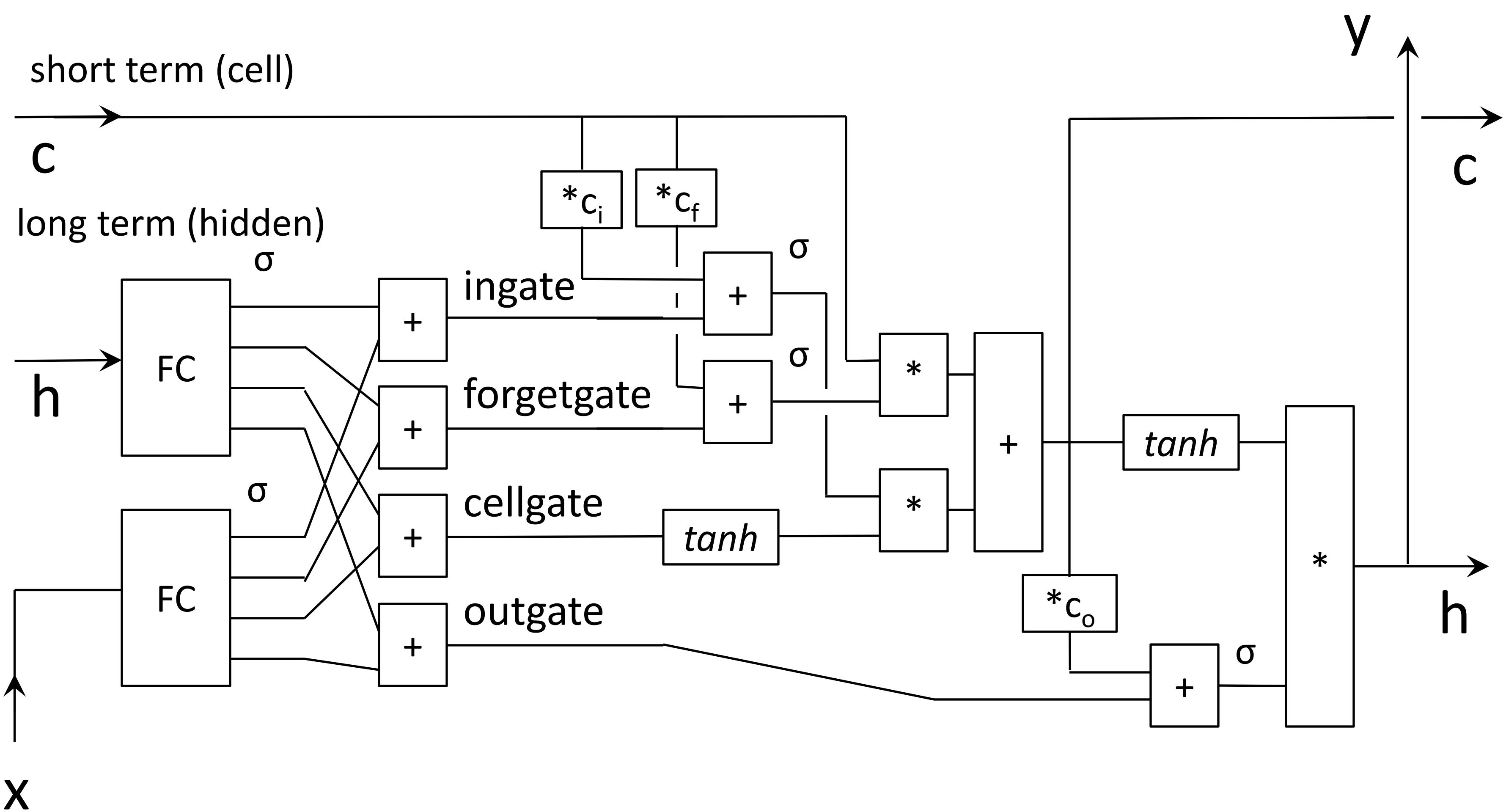


y
h
FC
σ
1-
*
*
+
h
x

Gated recurrent unit (GRU)



Long Short-Term Memory (LSTM)



Connectionist Temporal Classification (CTC)

- Chybová funkcia
- Počíta sumu pravdepodobností pre všetky možné zarovnania
- Zarovnanie je cesta cez tenzor pravdepodobností, ktorá pod odstránení BLANKu dáva požadovanú sekvenciu; pravdepodobnosť cesty je súčinom pravdepodobností krokov
- Počíta sa efektívne cez dynamické programovanie

Connectionist Temporal Classification (CTC)

- Ako vedľajší produkt vie vypočítať najpravdepodobnejšiu cestu, takže ju je možné použiť aj na dekódovanie pri inferencii (beam search)
- Po natrénovaní však celkom dobre funguje aj greedy: pre každý výstup sa vezme najpravdepodobnejší symbol
- Rozdiel: beam nájde cestu s najväčším súčinom pravdepodobností, greedy s najväčším súčtom