Computer Vision

Bogdan Alexe

bogdan.alexe@fmi.unibuc.ro

University of Bucharest, 2nd semester, 2023-2024

We

Bogdan
(Lecture)

Alexandra
(Lab classes)

You

407 (AI)

	Nume student	grupa
1	ALBU ELENA-ANGELICA	411
2	BARBU SEBASTIAN-MARIAN	411
3	DUTICA MARIA-DIANA	411
4	MOVILĂ GEORGE-DANIEL	411
5	NIȚĂ BOGDAN	411
6	PLĂIAN GEORGIANA	411
7	STOICA REMUS DANIEL	411
8	VASAI ANA-MARIA	411
9	GHIDĂNAC RUBEN-BENIAMIN	412

Programul de studii universitare de master ARTIFICIAL INTELLIGENCE
Durata studiilor: 2 ani (120 ECTS)
Forma de învățământ: cu frecvență

ANUL I 2023-2024 - PLAN DE INVATAMANT

(Sem.I - 14 săptămâni; Sem.II - 14 săptămâni) **Total Credite: 60**

Nr. Crt.	Discipline obligatorii	i	Semestrul I						Semestrul II					
			Tip oră				Forma de evaluare	Nr. de credite	Tip oră				Forma de evaluare	Nr. de credite
			C	S	L	P			C	S	L	P		
1.	Ob.11 **Învățare automata aplicată** / Practical Machine Learning	DS	2	1	-	-	V	6	-	-	-	-	-	-
2.	Ob.12 **Programare probabilistă** / Probabilistic Programming	DS	2	1	-	-	E	6	-	-	-	-	-	-
3.	Ob.13 **Reprezentarea cunoștințelor și inferență** / Knowledge representations and reasoning	DS	2	1	-	-	E	6	-	-	-	-	-	-
4.	Op.14 **Curs optional** / Optional Course	DS	2	1	-	-	E	6	-	-	-	-	-	-
5.	Ob.15 **Practică**/Practical Training	DS	-	2	-	-	V	4			-	-		
6.	Ob.16 **Deontologie Academică** / Academic Deontology	DC	1	-	-	-	V	2	-	-	-	-	-	-
7.	Ob.21 **Învățare automată** / Advanced machine learning	DS	-	-	-	-	-	-	2	1	-	-	E	6
8.	Ob.22 **Vedere Artificială** / Computer Vision	DS	-	-	-	-	-	-	2	1	-	-	E	6
9.	Ob.23 **Procesarea limbajului natural 1** / Natural language processing 1	DS	-	-	-	-	-	-	2	1	-	-	E	6
10.	Op.24 **Curs optional** / Optional Course	DS			-	-			2	1	-	-	E	6
11.	Ob.25 **Practică** / Practical Training	DS	-	-	-	-	-	-	-	2	-	-	V	6
Total			**9**	**6**	**-**	**-**	**3 E+ 3 V**	**30**	**8**	**6**	**-**	**-**	**4 E+ 1 V**	**30**

Schedule

INFO Master 407 (AI - Artificial Intelligence)

Universitatea din Bucuresti, Facultatea de Matematica si Informatica, str. Academiei 14, Bucuresti

	8 8:00 - 8:50	9 9:00 - 9:50	10 10:00 - 10:50	11 11:00 - 11:50	12 12:00 - 12:50	13 13:00 - 13:50	14 14:00 - 14:50	15 15:00 - 15:50	16 16:00 - 16:50	17 17:00 - 17:50	18 18:00 - 18:50	19 19:00 - 19:50
Lu Mon											Hristea F NatLangProc 1 (curs) 1(Stoilow)	
Ma Tue									Alexe B ComputerVision (curs) 2(Pompeiu)		Alexe B Gr_3 AdvMachLearn (sem, SI) 220 Alexe B Gr_4 AdvMachLearn (sem, SP) 220	
Mi Wed	Alexe B AdvMachLearn (curs) 1(Stoilow)		Alexe B Gr_1 AdvMachLearn (sem, SI) 219 Ciocan I Gr_2 NatLangProc 1 (Lab, SI) L-308 Ciocan I Gr_3 NatLangProc 1 (Lab, SP) L-308 Alexe B Gr_2 AdvMachLearn (sem, SP) 219									
Jo Thu			Uban A Gr_1 NatLangProc 1 (Lab, SI) L-308								Diaconu Al Gr_1 ComputerVision (Lab, SI) L-303 Diaconu Al Gr_3 ComputerVision (Lab, SP) L-303	
Vi Fri	Diaconu Al Gr_2 ComputerVision (Lab, SI) L-303 Diaconu Al Gr_4 ComputerVision (Lab, SP) L-303											

Orar generat:25.02.2024

aSc Orare

Schedule

8	Ob.22 **Vedere Artificială** Computer Vision	-	-	-	-	2	1	E	6

- 2 hours lecture each week (14 lectures in total)
- 2 hours lab session every two weeks (7 lab sessions in total)

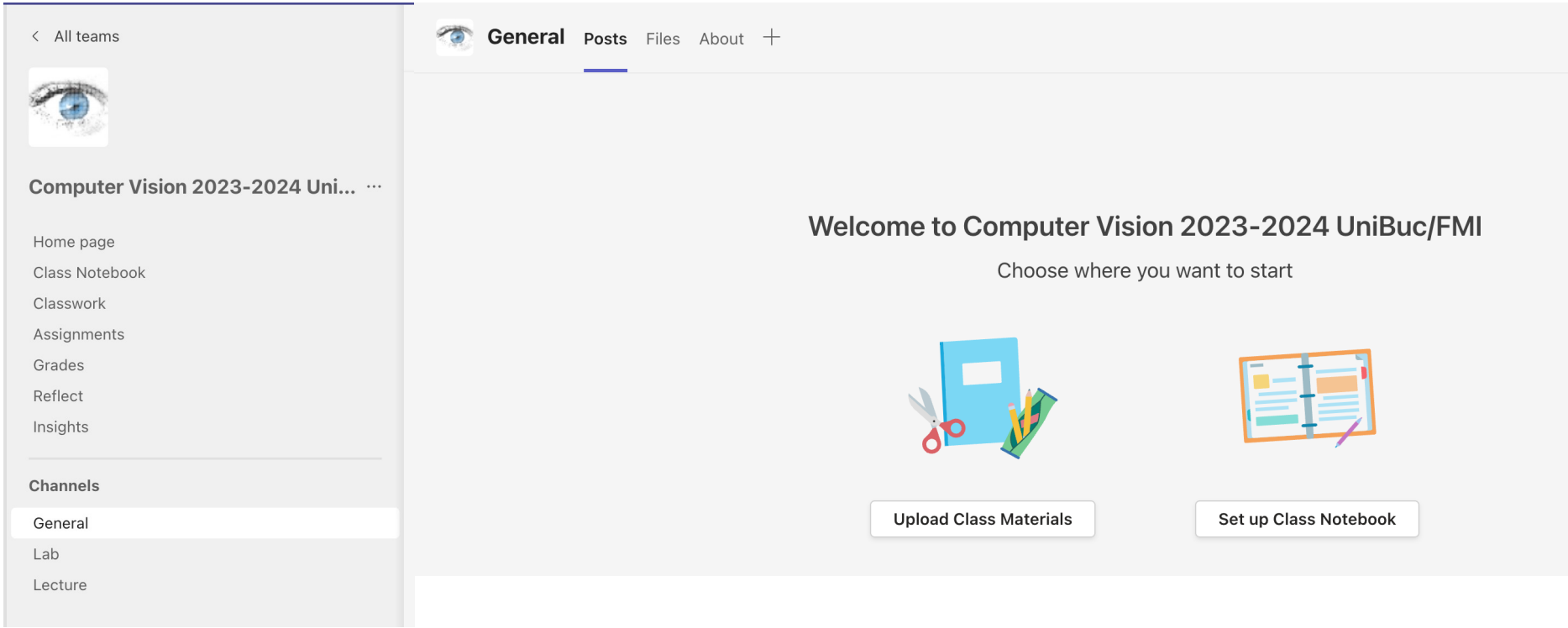
- Lecture: Tuesday, 16-18 (Bogdan), amf. Pompeiu, weekly

- Lab (one group every two weeks):
  - Thursday 18-20, room 303 - Alexandra (g1 and g3)
  - Friday 8-10, room 303 - Alexandra (g2 and g4)

"10000-hour rule" - the key to achieving true expertise in any skill is simply a matter of practicing, albeit in the correct way, for at least 10 000 hours (40h/week × 50 weeks × 5 years)

F2F vs online

- All lectures and lab sessions are face to face
- We will put the materials for lecture and lab sessions in TEAMS.



Exam – evaluation in June

Grade = min(10, P1 + P2 + bonus)

- P1 = project 1 = 5 points (+ some bonus?)
- P2 = project 2 = 5 points (+ some bonus?)
- bonus = discuss later
- no constraints, with 4.99 you fail, with 5 you pass

Exam – evaluation in July (restanță)

Grade = min(10, P3 + max(P1, P2, P3) + bonus)

- P3 = project 3 = 5 points
- bonus = discuss later
- no constraints, with 4.99 you fail, with 5 you pass

Exam – evaluation in September (reexaminare)

Grade = min(10, P4 + max(P1, P2, P4) + bonus)

- P4 = project 4 = 5 points
- bonus = discuss later
- no constraints, with 4.99 you fail, with 5 you pass

Exam – evaluation in September (mărire)

Grade = min(10, 2 × P4 + bonus)

- P4 = project 4 = 5 points
- bonus = discuss later

Bonus

- maximum 1 point that you can get as a bonus added to your grade
- only available in this academic year
- Bonus = lecture bonus + lab session bonus
- Lecture bonus (quizes on Kahoot!):
  - at the end of each lecture you will get a few questions from that lecture. Answer correct and fast to be higher in ranking.
  - we will use Kahoot!
  - based on the Kahoot! ranking at each lecture we will award points
    - students ranked 1-3 get 0.15p
    - students ranked 4-6 get 0.10p
    - students ranked 7-9 get 0.05p
- Lab session bonus (lab attendance):
  - you get 0.1 points per each lab session that you attend (maximum 0.1 points every two weeks). As there are 7 lab sessions you could earn at most 0.7 points.

Kahoot! - test

- use your phone/laptop to open www.kahoot.com and select PLAY
- use the code given by me
- choose your username, ideally based on your real name (bogdan.alexe)
- **read the question, answer on your phone/laptop selecting the right answer**
- quick test!

Resources

github.com/jbhuang0604/awesome-computer-vision/

Awesome Computer Vision: 

A curated list of awesome computer vision resources, inspired by awesome-php.

For a list people in computer vision listed with their academic genealogy, please visit here

Contributing

Please feel free to send me pull requests or email (jbhuang@vt.edu) to add links.

Table of Contents

Books

Computer Vision

- Computer Vision: Models, Learning, and Inference - Simon J. D. Prince 2012
- Computer Vision: Theory and Application - Rick Szeliski 2010
- Computer Vision: A Modern Approach (2nd edition) - David Forsyth and Jean Ponce 2011
- Multiple View Geometry in Computer Vision - Richard Hartley and Andrew Zisserman 2004
- Computer Vision - Linda G. Shapiro 2001
- Vision Science: Photons to Phenomenology - Stephen E. Palmer 1999
- Visual Object Recognition synthesis lecture - Kristen Grauman and Bastian Leibe 2011
- Computer Vision for Visual Effects - Richard J. Radke, 2012
- High dynamic range imaging: acquisition, display, and image-based lighting - Reinhard, E., Heidrich, W., Debevec, P., Pattanaik, S., Ward, G., Myszkowski, K 2010
- Numerical Algorithms: Methods for Computer Vision, Machine Learning, and Graphics - Justin Solomon 2015

OpenCV Programming

- Learning OpenCV: Computer Vision with the OpenCV Library - Gary Bradski and Adrian Kaehler
- Practical Python and OpenCV - Adrian Rosebrock
- OpenCV Essentials - Oscar Deniz Suarez, Mª del Milagro Fernandez Carrobles, Noelia Vallez Enano, Gloria Bueno Garcia, Ismael Serrano Gracia

Courses

Computer Vision

- EENG 512 / CSCI 512 - Computer Vision - William Hoff (Colorado School of Mines)
- Visual Object and Activity Recognition - Alexei A. Efros and Trevor Darrell (UC Berkeley)
- Computer Vision - Steve Seitz (University of Washington)
- Visual Recognition Spring 2016, Fall 2016 - Kristen Grauman (UT Austin)
- Language and Vision - Tamara Berg (UNC Chapel Hill)
- Convolutional Neural Networks for Visual Recognition - Fei-Fei Li and Andrej Karpathy (Stanford University)
- Computer Vision - Rob Fergus (NYU)
- Computer Vision - Derek Hoiem (UIUC)
- Computer Vision: Foundations and Applications - Kalanit Grill-Spector and Fei-Fei Li (Stanford University)
- High-Level Vision: Behaviors, Neurons and Computational Models - Fei-Fei Li (Stanford University)
- Advances in Computer Vision - Antonio Torralba and Bill Freeman (MIT)
- Computer Vision - Bastian Leibe (RWTH Aachen University)
- Computer Vision 2 - Bastian Leibe (RWTH Aachen University)
- Computer Vision Pascal Fua (EPFL):
- Computer Vision 1 Carsten Rother (TU Dresden):
- Computer Vision 2 Carsten Rother (TU Dresden):
- Multiple View Geometry Daniel Cremers (TU Munich):

What is Computer Vision?

Are we done?

Slide credit K. Graumann

What is Computer Vision?

Make computers understand images and video.

What kind of scene?

Where are the cars?

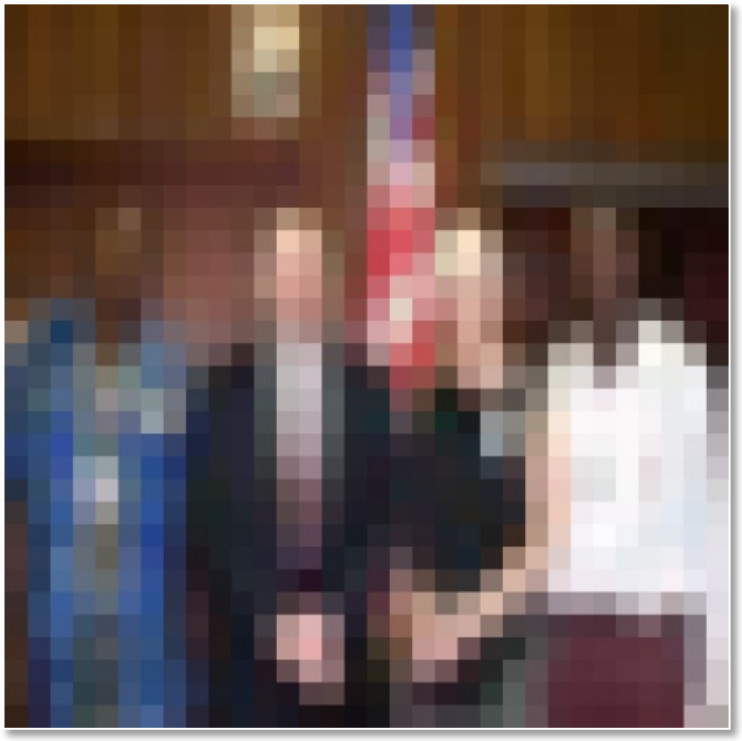
How far is the building?

…

Slide credit D. Hoiem

The goal of Computer Vision

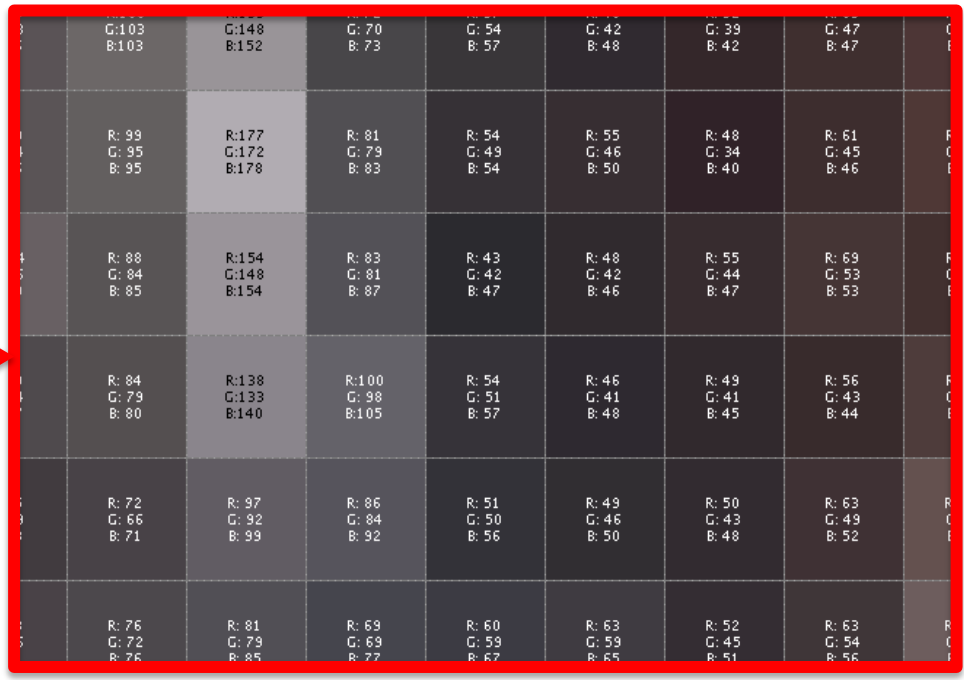
- Extract information from pixels



Slide credit S. Lazebnik

The goal of Computer Vision

- Extract information from pixels



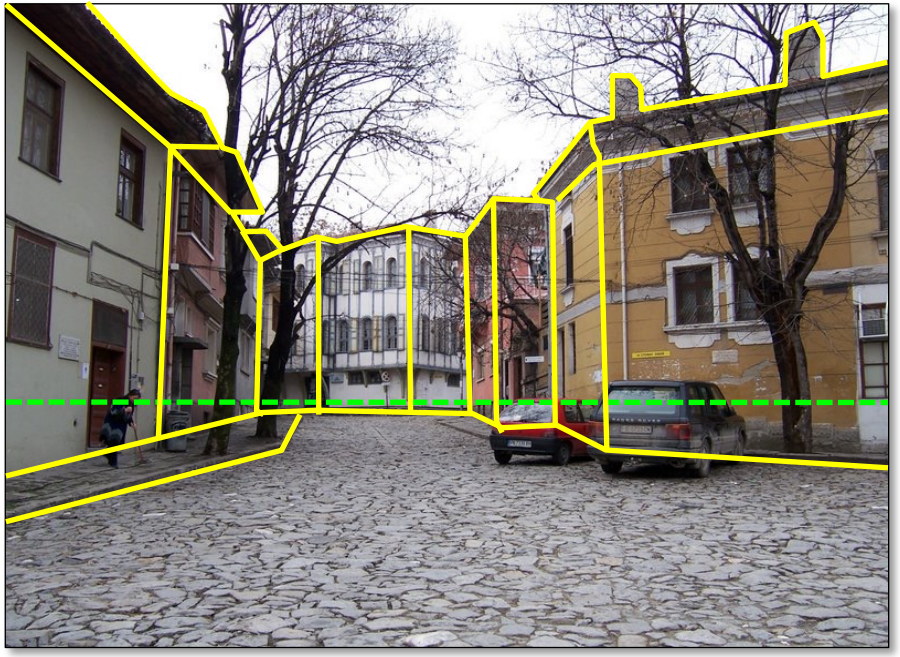
Human vision

Computer vision

Slide credit S. Lazebnik

What kind of information?

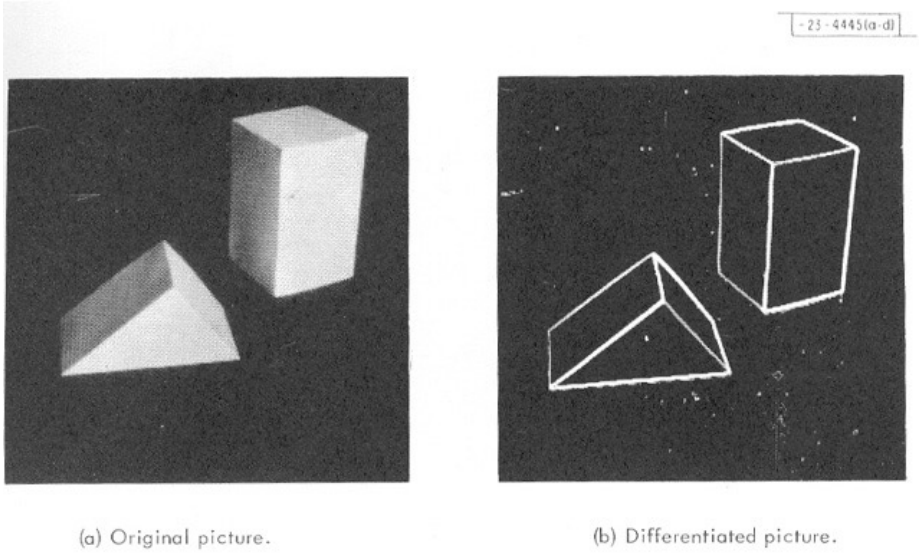
Semantic information



Geometric information (3D)

Slide credit S. Lazebnik

Visual data in 1963



(a) Original picture.

(b) Differentiated picture.

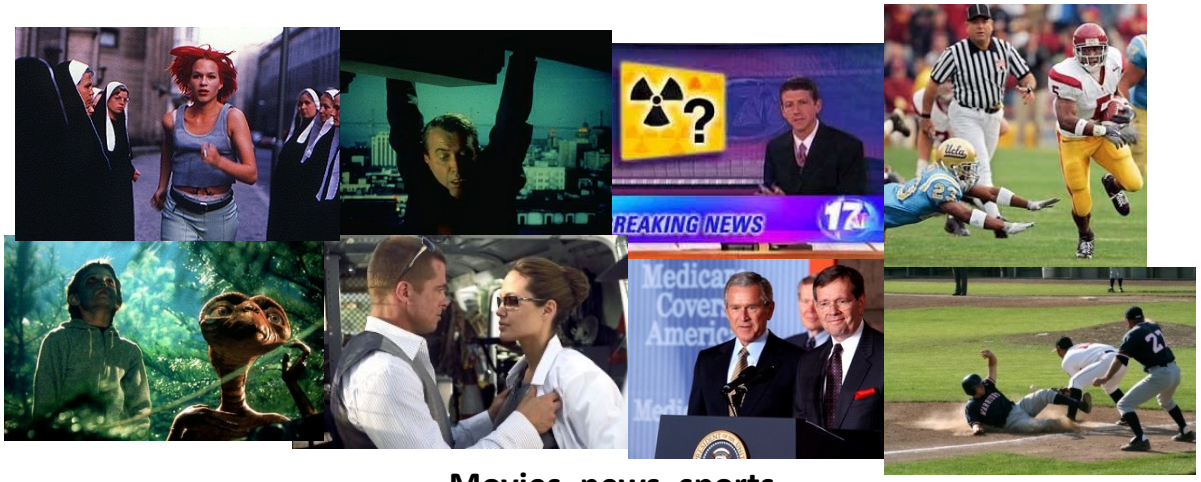
(c) Line drawing.

(d) Rotated view.

L. G. Roberts *Machine Perception of Three Dimensional Solids*,
Doctoral thesis, MIT, 1963.

Visual data in 2024

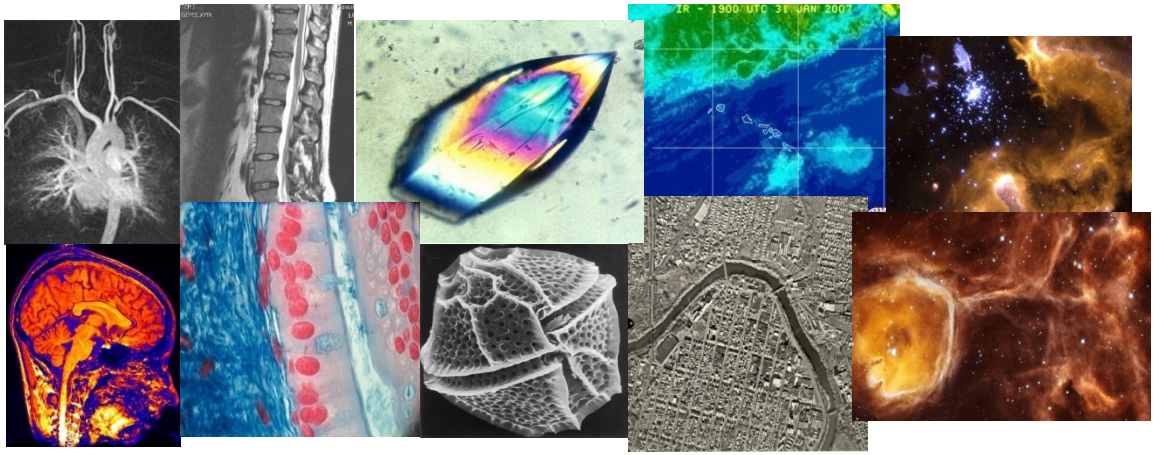
Photo albums



Movies, news, sports

Google Image Search™ Picasa™ flickr™ facebook twitter YouTube Broadcast Yourself™

Video surveillance and security



Medical imaging

Slide credit S. Lazebnik

Successful applications in Computer Vision

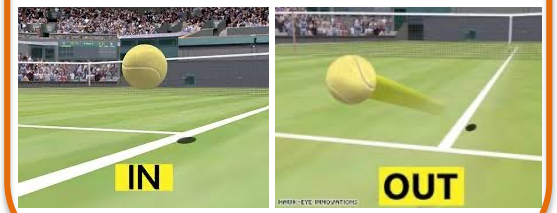
OCR

Xbox Kinect

Visual search

Face detection / recognition

Hawk-eye decision system



Driving assistance systems

Course structure

1. **Features and filters: low-level vision**

   Linear filters, color, texture, edge detection

2. **Grouping and fitting: mid-level vision**

   Fitting curves and lines, robust fitting, RANSAC, Hough transform, segmentation

3. **Multiple views**

   Local invariant feature and description, epipolar geometry and stereo, object instance recognition

4. **Object Recognition: high – level vision**

   Object classification, object detection, part based models, bovw models

5. **Video understanding**

   Object tracking, background subtraction, motion descriptors, optical flow

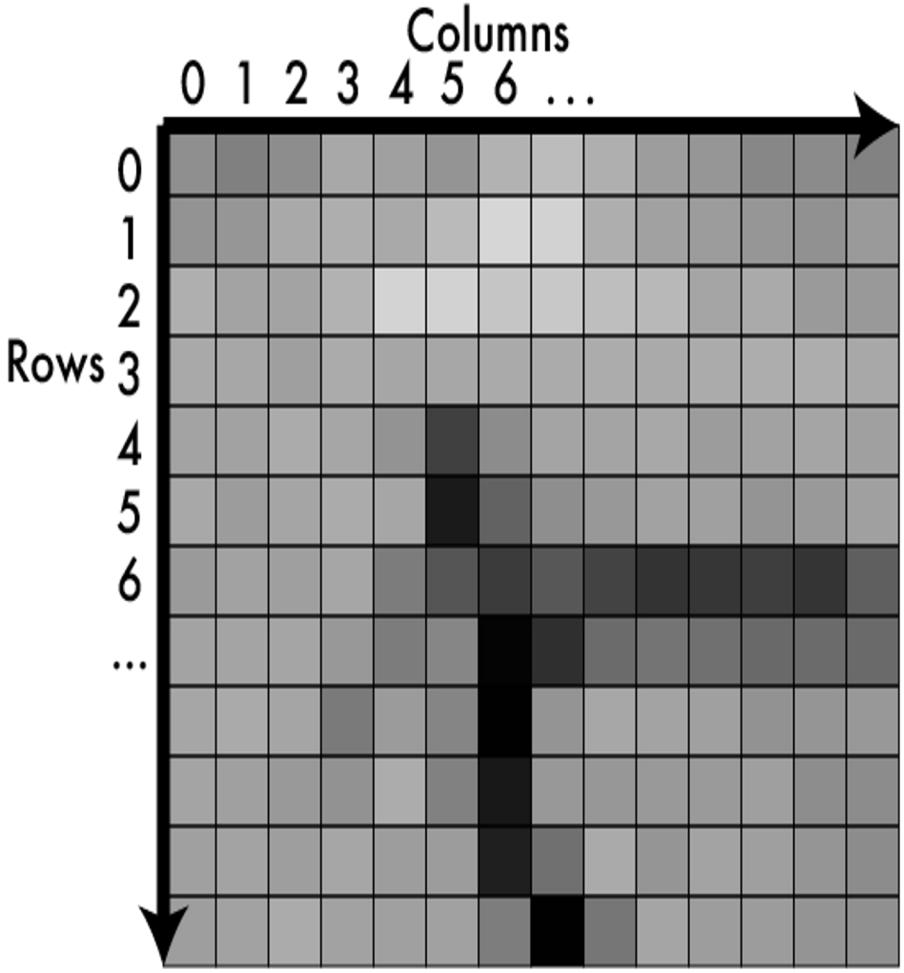
Computer Vision - Laboratory class 1

Introduction to Computer Vision

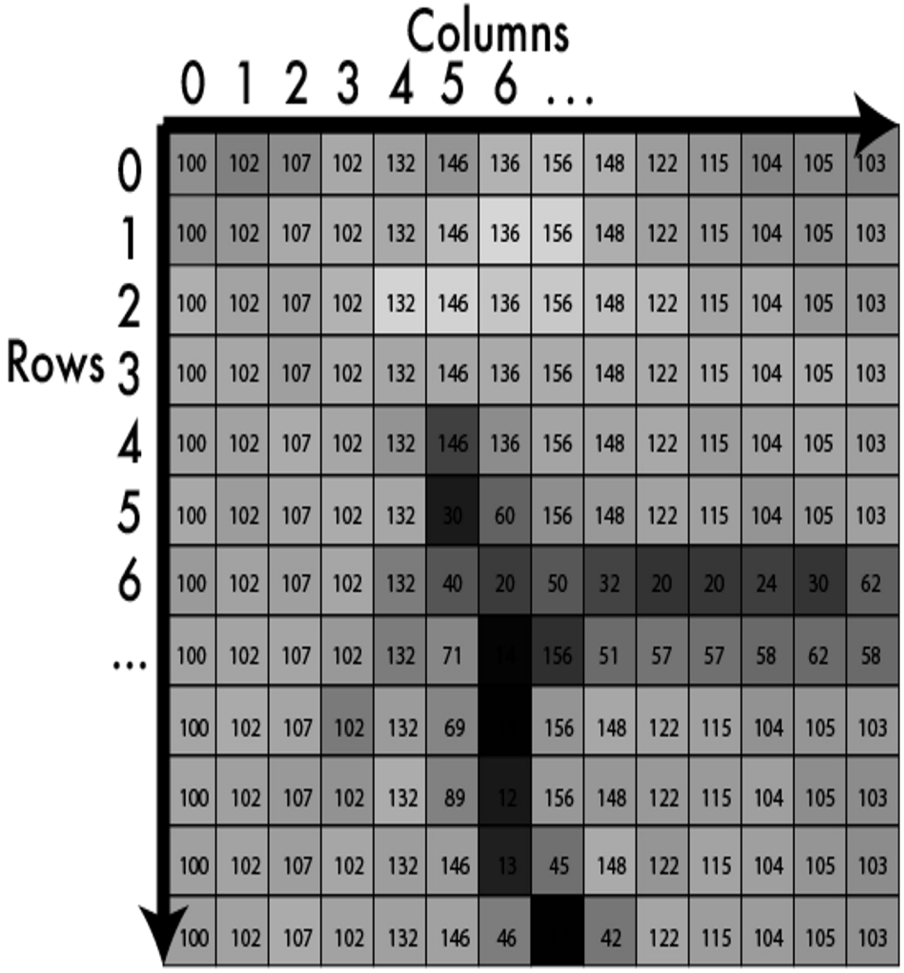
1.1 Introduction to Computer Vision using OpenCV

In the first part of this laboratory class we will introduce the OpenCV library in Python. During the next laboratory classes we will rely on OpenCV for solving different exercises. The script *Lab1-intro-opencv.ipynb* contains basic information regarding installing the OpenCV library and using several functions from this library for loading and displaying an image, accessing individual pixels, array slicing and cropping, resizing images, rotating an image, etc.

An image is a matrix of light

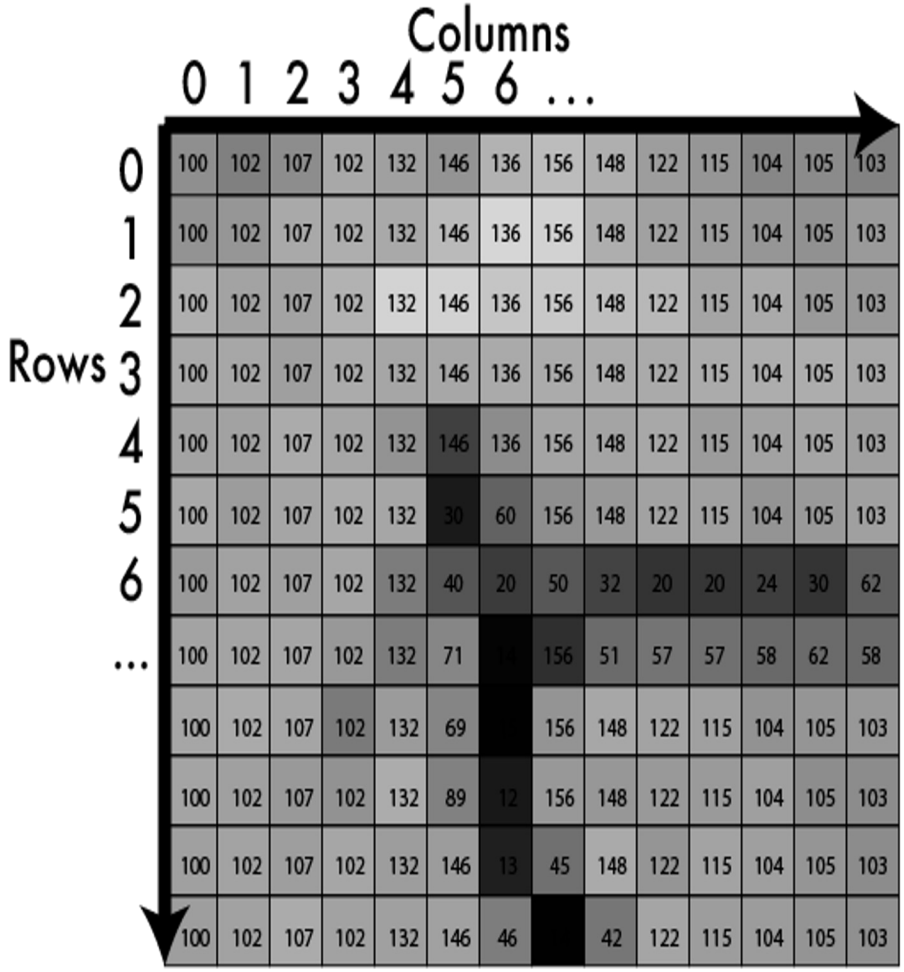


Values in matrix = how much light

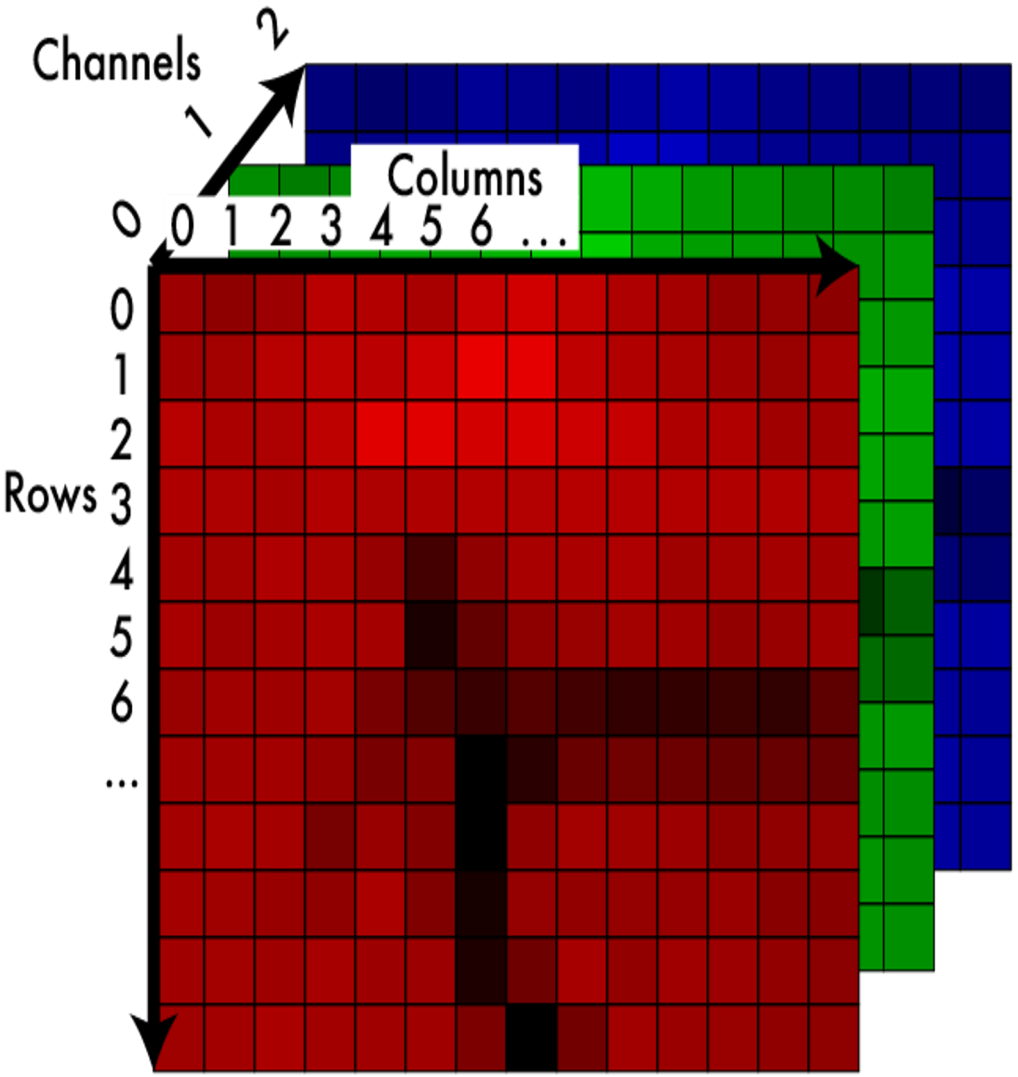


Values in matrix = how much light

- Higher = more light
- Lower = less light
- Bounded
  - No light = 0
  - Sensor/device limit = max
  - Typical ranges:
    - [0-255], fit into a byte
    - [0-1], floating point
- Called pixels

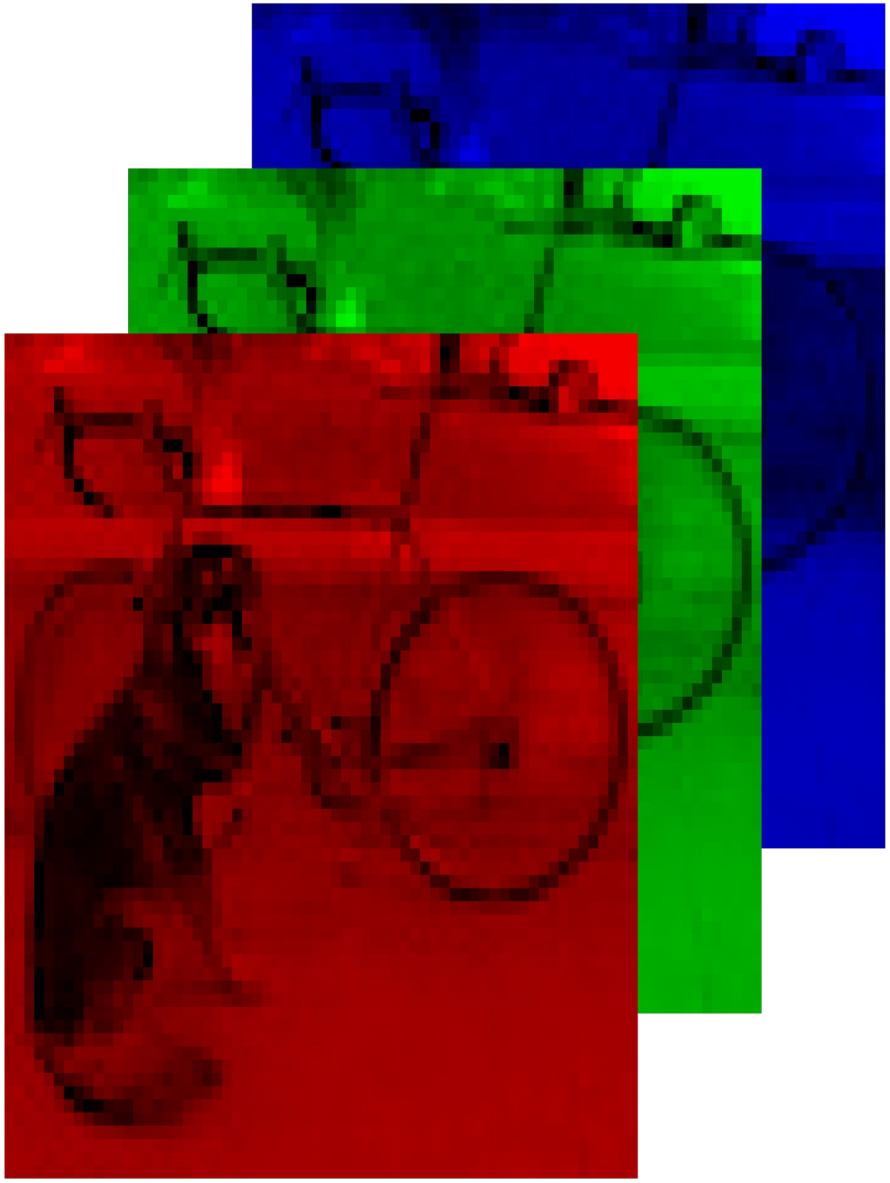


Color image: 3d tensor in colorspace



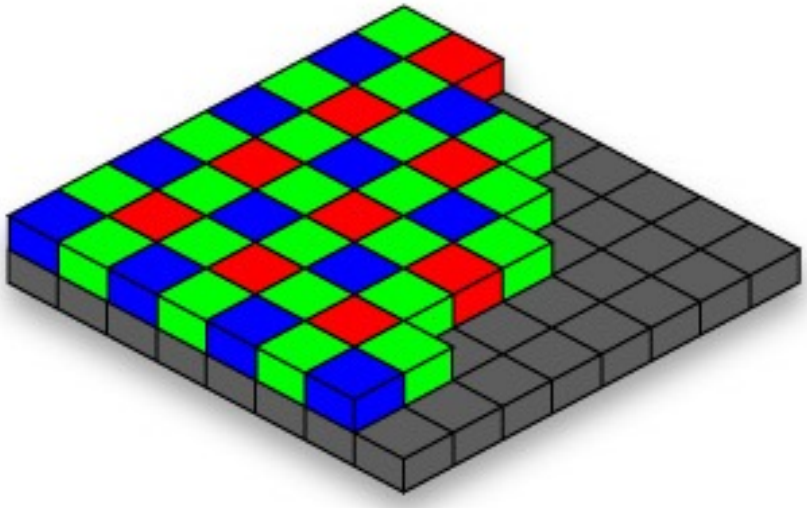
RGB information in separate “channels”

- each channel encodes one primary, use RGB for now.
- we can match “real” colors using a mix of primaries.
- adding the light produced from each primary mimics the original color.

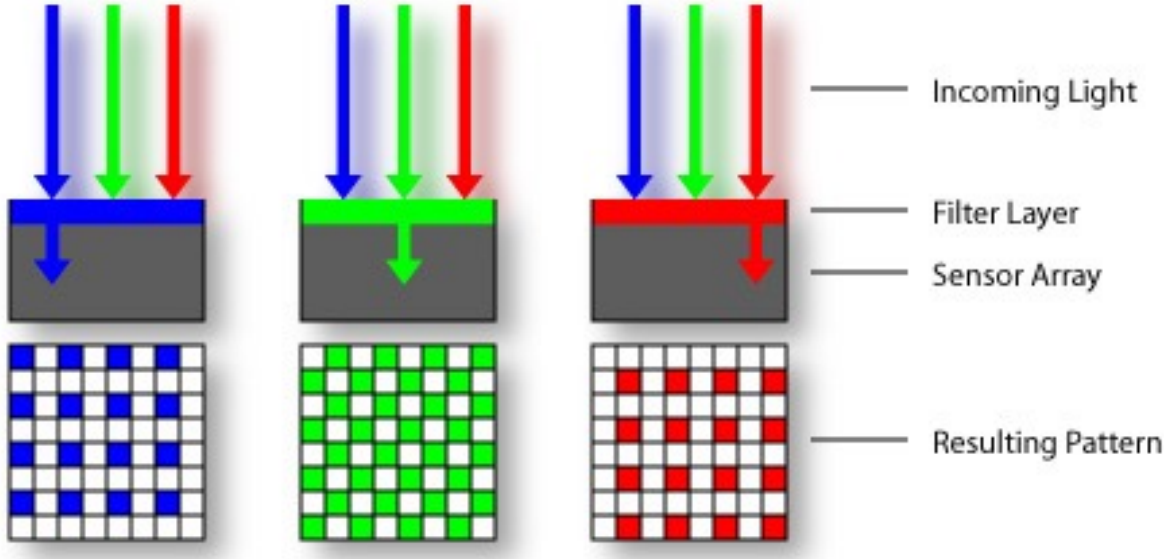


First laboratory class

Color Sensing: Bayer Grid



Estimate RGB at each cell from neighboring values

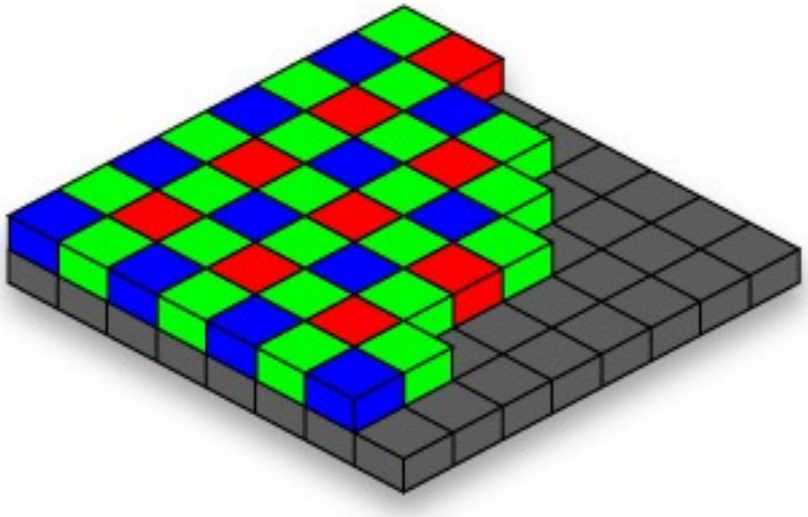


http://en.wikipedia.org/wiki/Bayer_filter

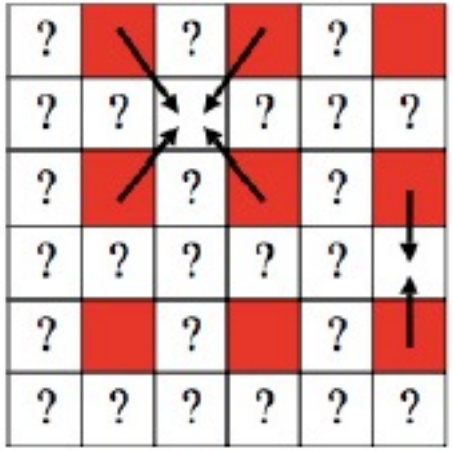
Slide credit Steve Seitz

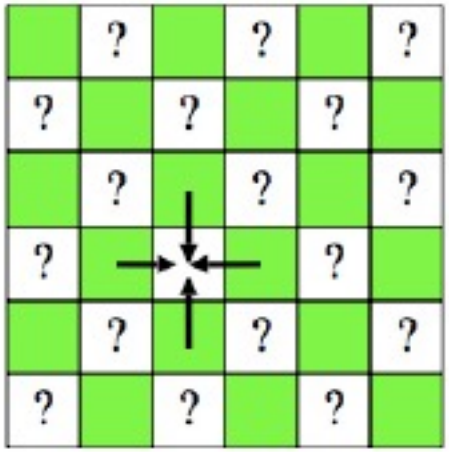
First laboratory class

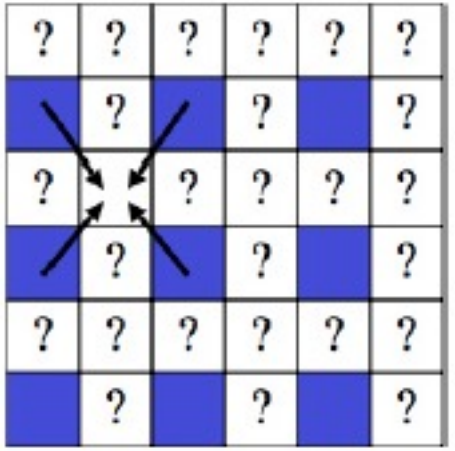
Color Sensing: Bayer Grid



Estimate RGB at each cell from neighboring values







Computer Vision - Laboratory class 1

Introduction to Computer Vision

1.3 Demosaicing

In this exercise, we are going to 'demosaic' an image encoded with the Bayer Pattern. There are some cameras that use the Bayer Pattern in order to save an image. Using this encoding only 50% of green pixels, 25% of red pixels and 25% of blue pixels are kept. The Bayer encoding takes a RGB image and encodes it in one of the four possible ways as depicted in Figure 2.

Figure 2: *Bayer pattern, specified as one of the four possible values. Each value represents the order of the red, green, and blue sensors by describing the four pixels in the upper-left corner of the image (left-to-right, top-to-bottom).*

In this exercise, we are going to 'demosaic' an encoded image assuming the encoding with the RGGB pattern. We will implement a very simple algorithm which, for each pixel, fills in the two missing channels by averaging the values of their nearest neighbors (1, 2 or 4)

Computer Vision - Laboratory class 1

Introduction to Computer Vision

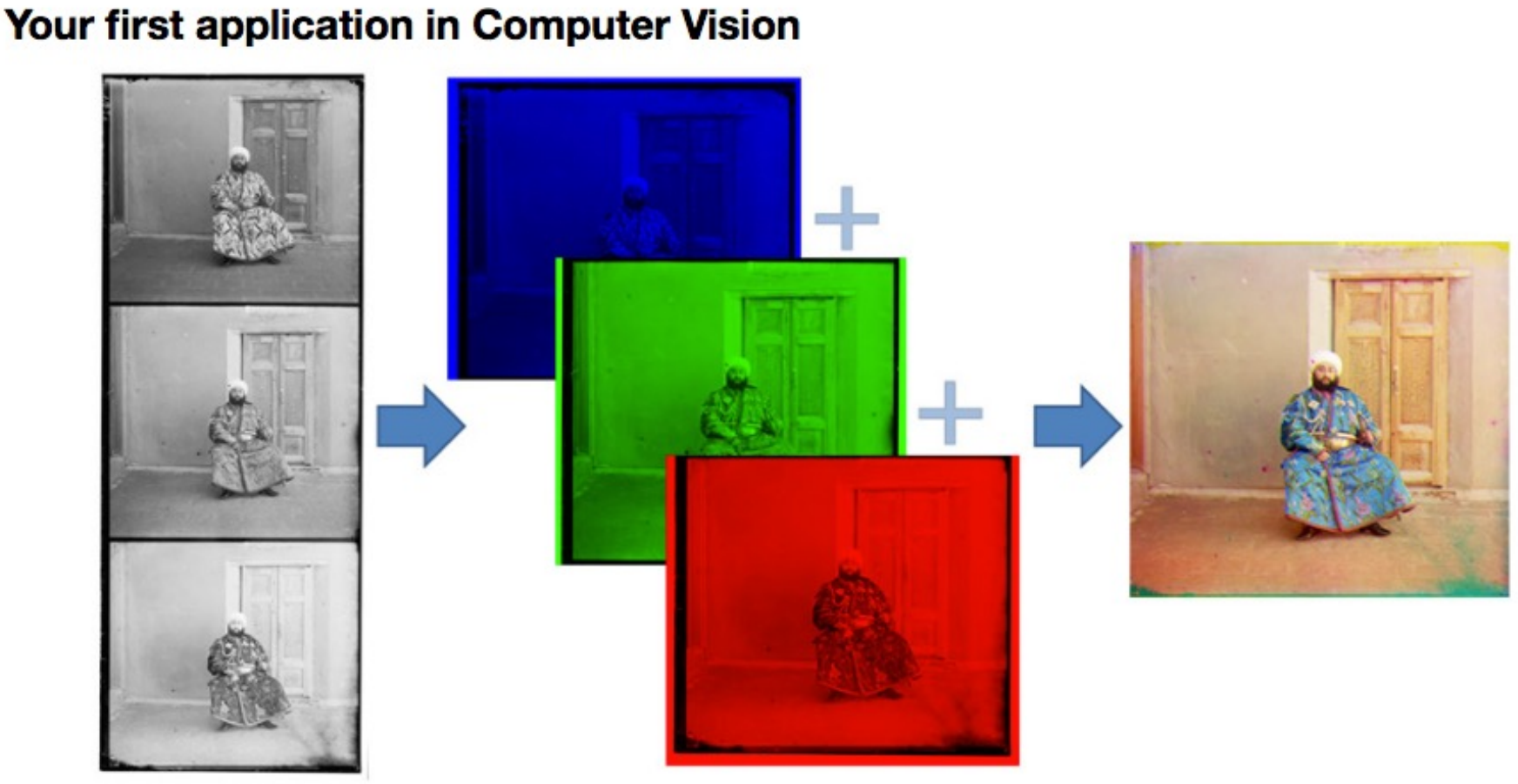
1.3 Demosaicing

demosaicing

Image encoded with the Bayer pattern RGGB

corresponding color image

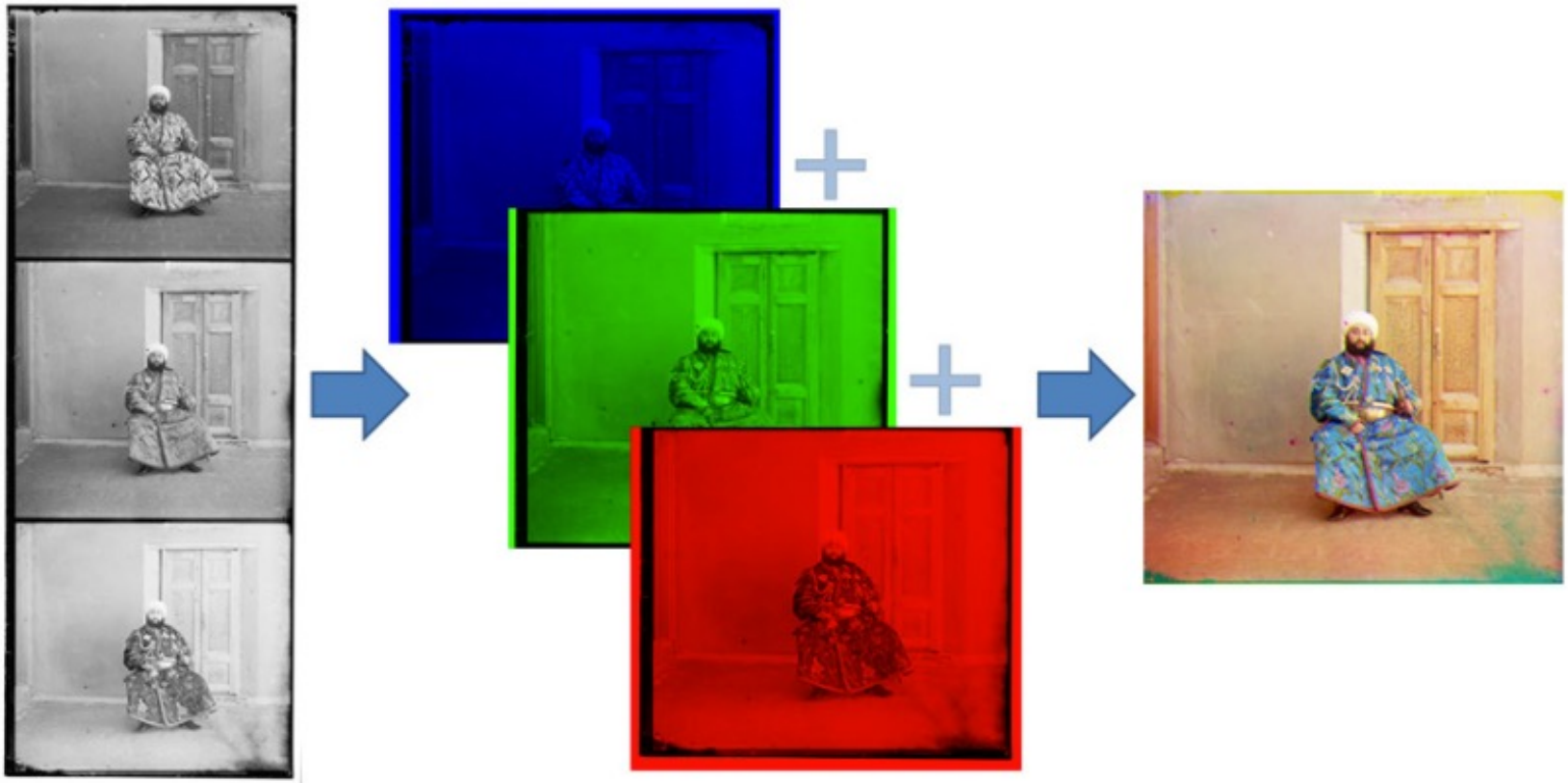
First laboratory class



Record three exposures of every scene onto a glass plate using a red, a green, and a blue filter and then project the monochrome pictures with correctly colored light to reproduce the color image

First laboratory class

Your first application in Computer Vision



Your program will take a **glass plate image** as input and produce a **single color image** as output. The program should *divide the image into three equal parts* and **align the second and the third parts (G and R) to the first (B)**.

Not aligned

Aligned

Computer Vision - Laboratory class 1

Introduction to Computer Vision

1.2 Aligning color channels from digitized glass plate images

Sergei Mikhailovich Prokudin-Gorskii (1863-1944) was a photographer ahead of his time. He saw color photography as the wave of the future and came up with a simple idea to produce color photos: record three exposures of every scene onto a glass plate using a red, a green, and a blue filter and then project the monochrome pictures with correctly coloured light to reproduce the color image; color printing of photos was very difficult at the time. Due to the fame he received from his color photos, including the only color portrait of Leo Tolstoy (a famous Russian author), he won the Tzar's permission and funding to travel across the Russian Empire and document it in 'color' photographs. His RGB glass plate negatives were purchased in 1948 by the Library of Congress. They are now digitized and available online.

Computer Vision projects for undergraduate students

1. Context aware image resizing

Initial image

Resize the image:
increase/decrease width/height of the image

Main idea: add/remove
irregularly shaped “seams”

How to chose?

[Shai & Avidan, SIGGRAPH 2007]

Slide credit K. Grauman

Context aware image resizing

Context aware resizing

Traditional resizing

[Shai & Avidan, SIGGRAPH 2007]

Slide credit K. Grauman

Main idea: Context aware image resizing

Context aware resizing

Intuition:

- Preserve the most "interesting" content
  → Prefer to remove pixels with low gradient energy
- To reduce or increase size in one dimension, remove or add irregularly shaped "seams"
  → Optimal solution via dynamic programming.

Slide credit K. Grauman

Main idea: seam carving

image I

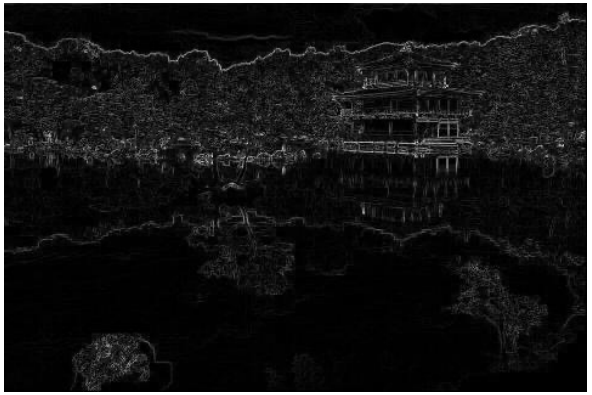


Image gradient

$$\nabla I = \sqrt{(\frac{\partial I}{\partial x})^2 + (\frac{\partial I}{\partial y})^2}$$

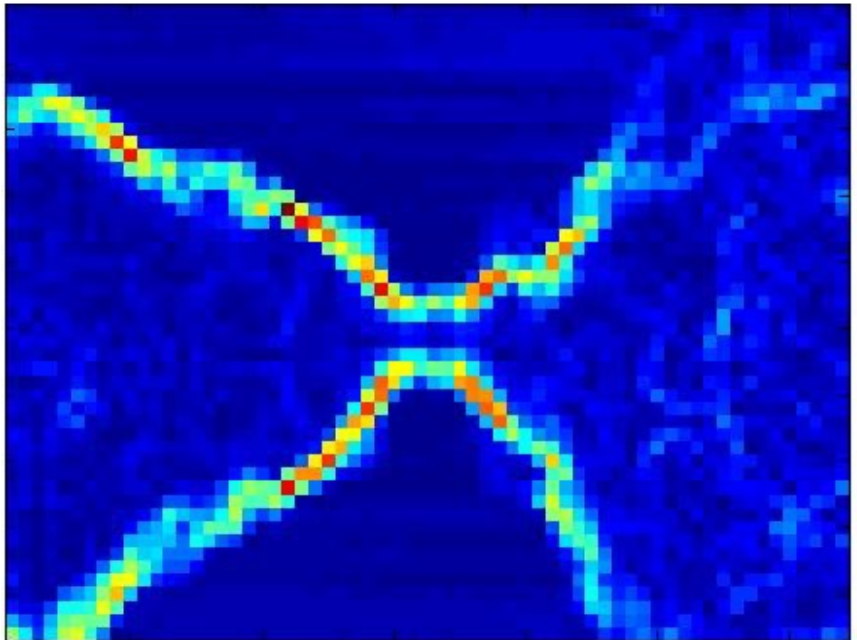
- Want to remove seams where they won't be very noticeable:
  - measure "energy" as gradient magnitude
- Choose seam based on **minimum total energy path** across image, subject to 8-connectedness.

Slide credit K. Grauman

Real image example

Original Image

Energy Map



Blue = low energy

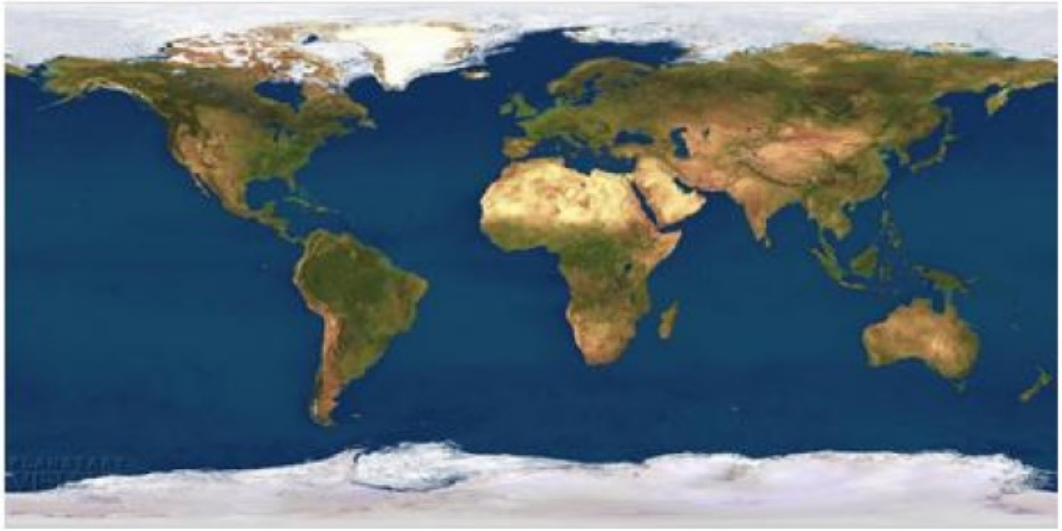
Red = high energy

Slide credit K. Grauman

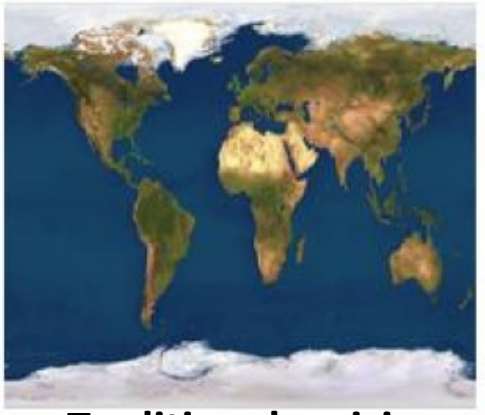
Real image example

Slide credit K. Grauman

Results



Context aware image resizing



Traditional resizing

Slide credit K. Grauman

“Failure cases” with seam carving

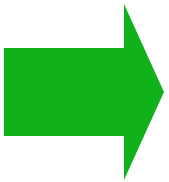
Slide credit K. Grauman

Removal of a marked object

Slide credit K. Grauman

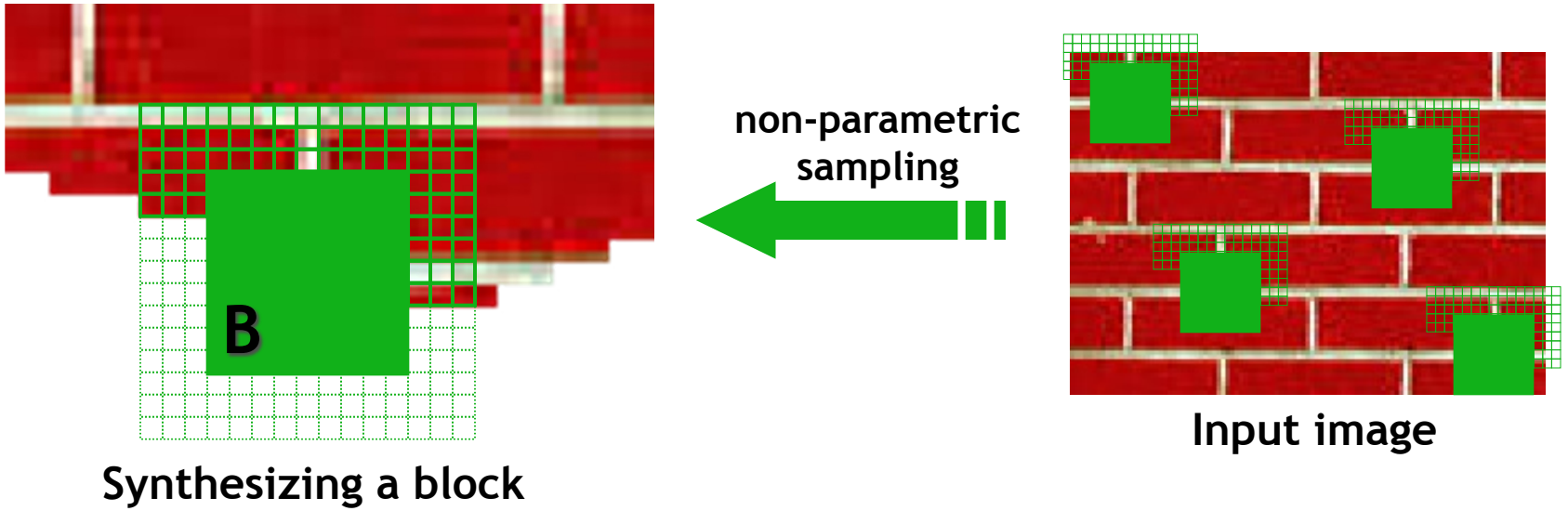
2. Texture synthesis

- Goal: create new samples of a given texture
- Many applications: virtual environments, hole-filling, texturing surfaces



Slide credit A. Efros

Image Quilting [Efros & Freeman 2001]

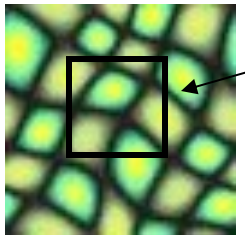


- Observation: neighbor pixels are highly correlated

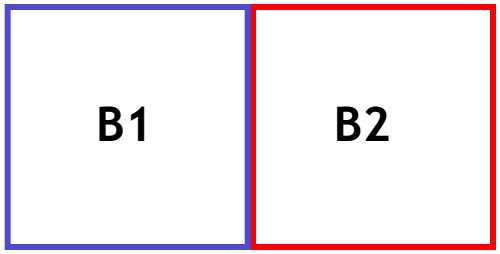
Idea: unit of synthesis = block

- Exactly the same but now we want P(**B**|N(**B**))
- Much faster: synthesize all pixels in a block at once

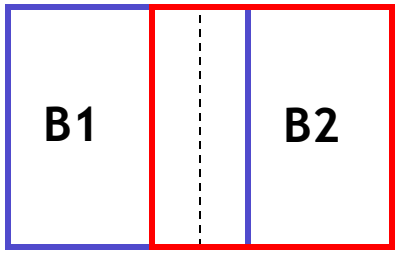
Slide credit A. Efros



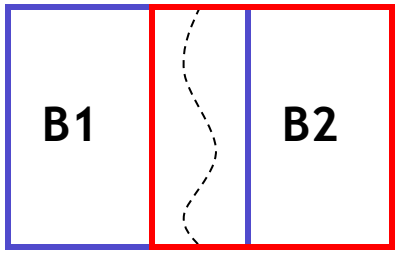
Input texture



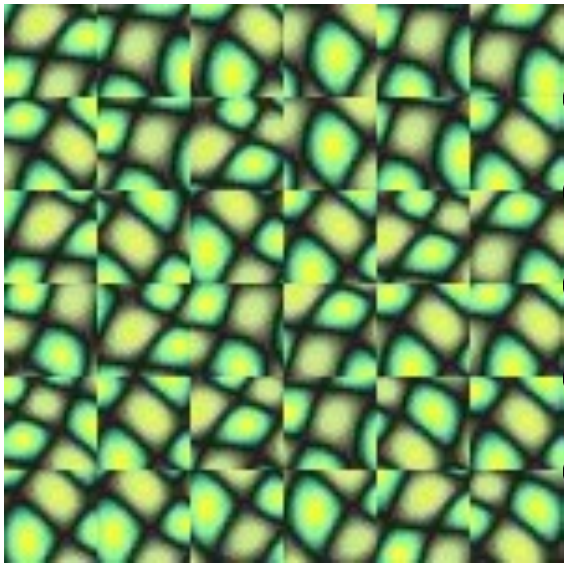
Random placement of blocks

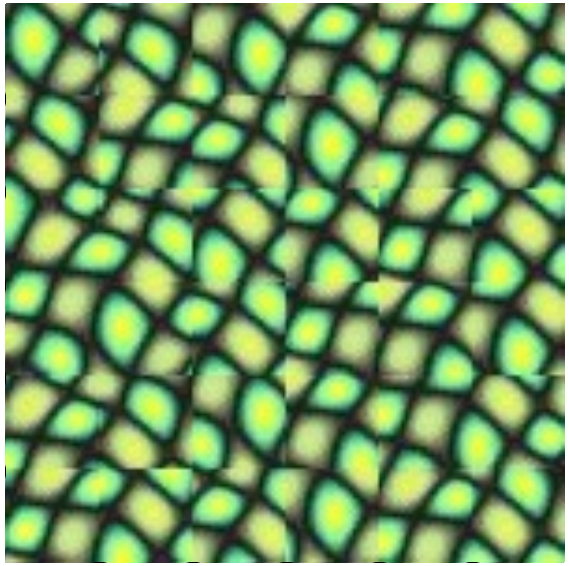


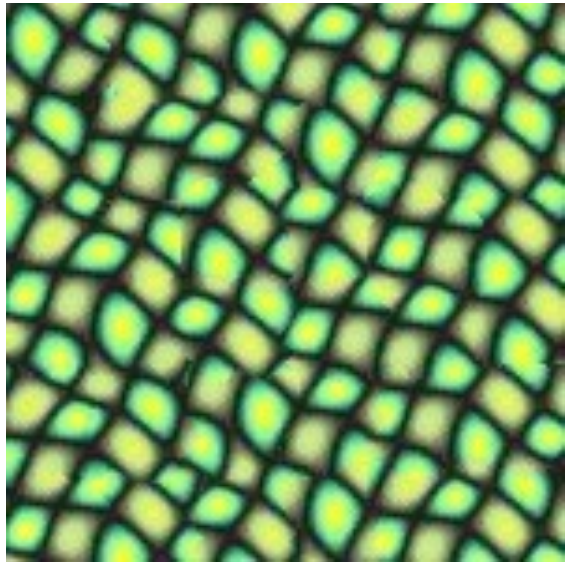
Neighboring blocks constrained by overlap



Minimal error boundary cut

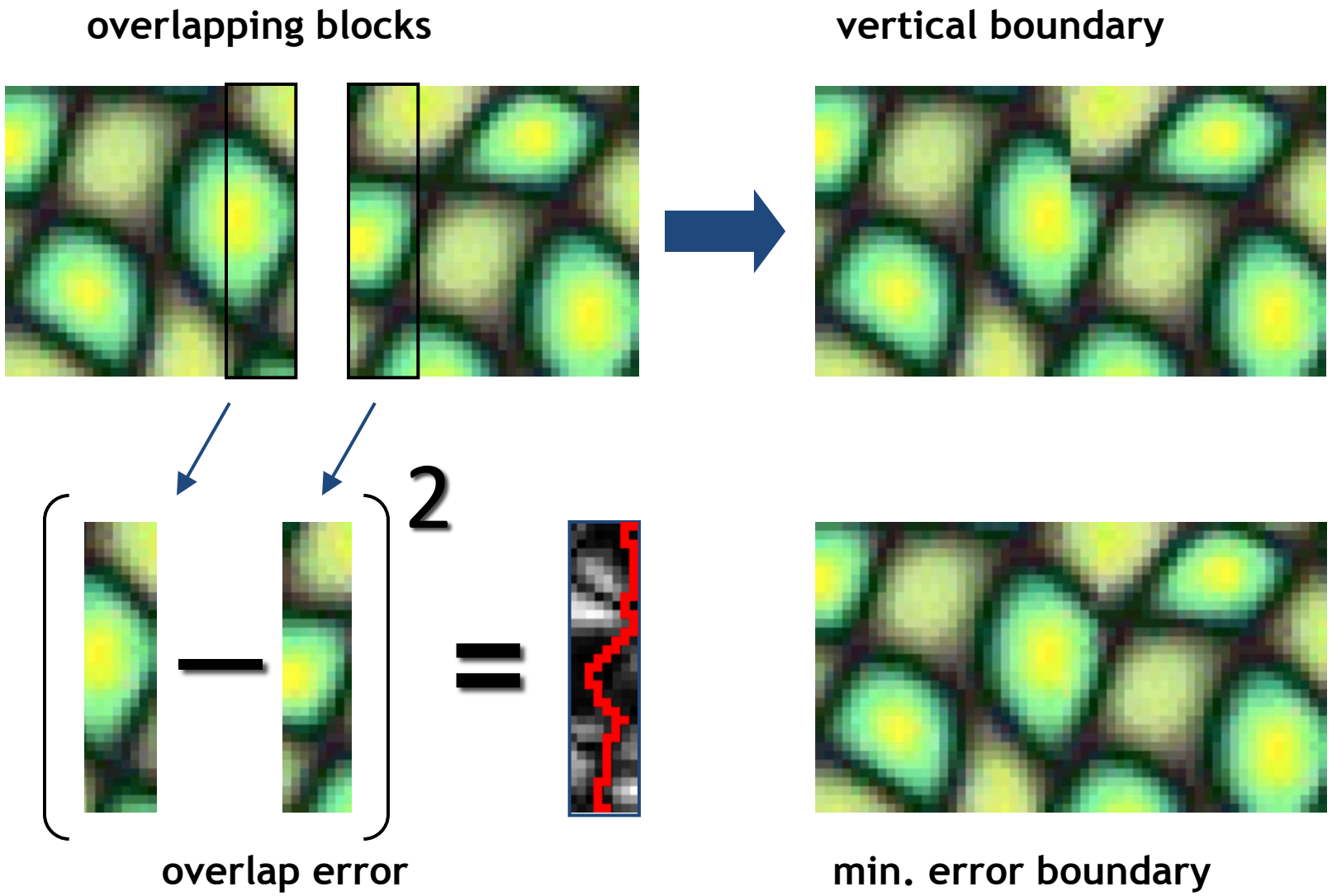




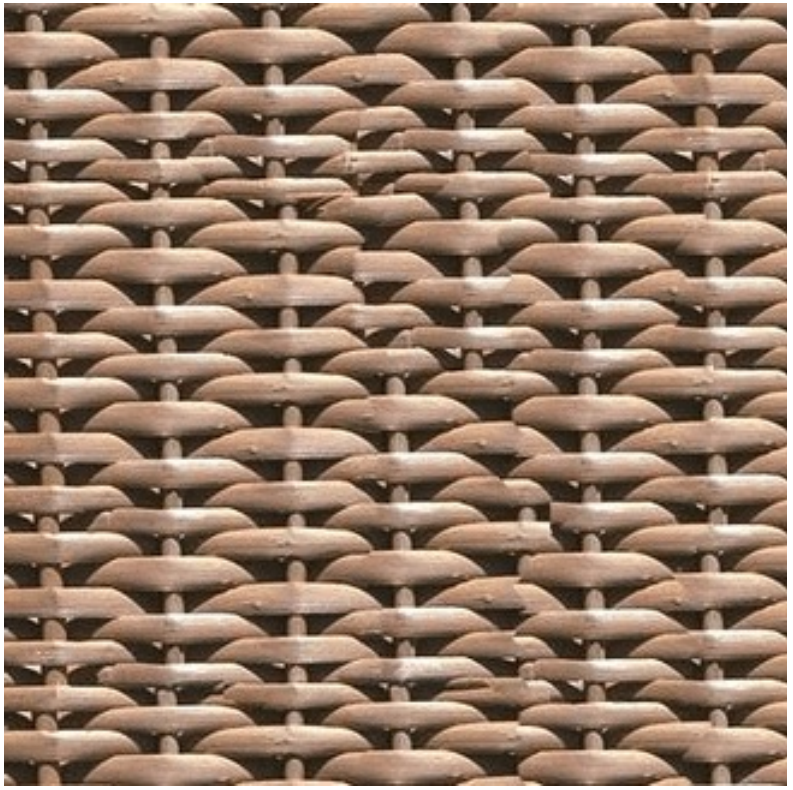


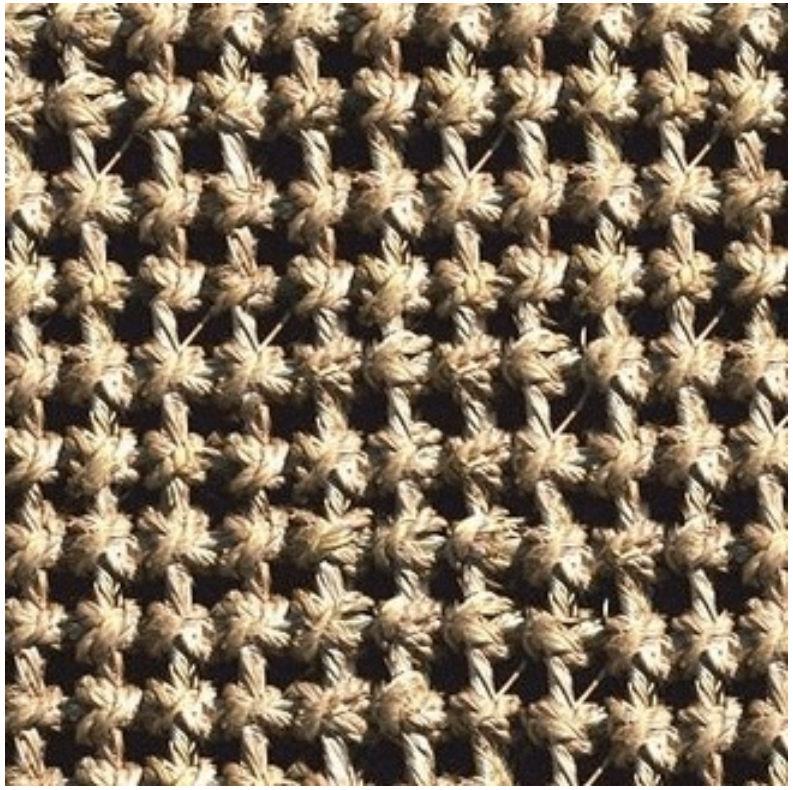
Slide credit A. Efros

Minimal error boundary



Slide credit A. Efros



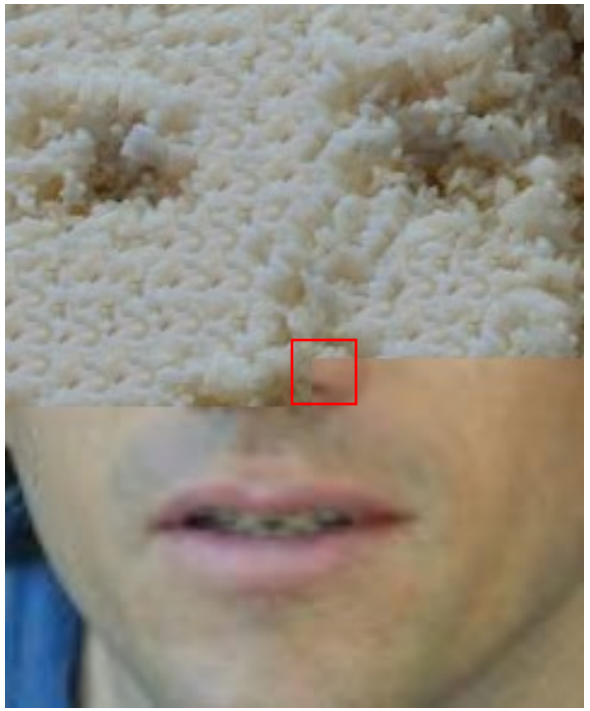



College Inn
Chicken Broth


College Inn
Chicken Broth

Texture Transfer

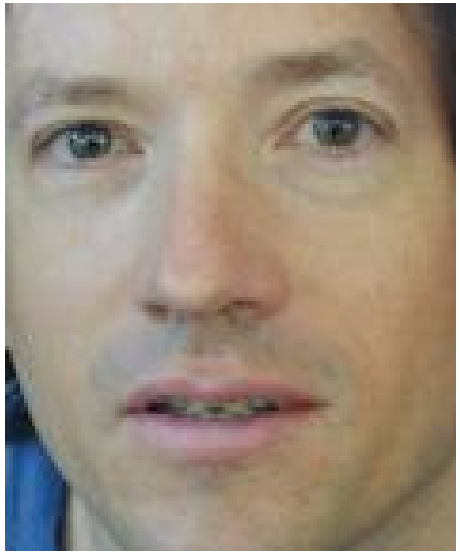
- Take the texture from one object and “paint” it onto another object
  - this requires separating texture and shape
  - that’s HARD, but we can cheat
  - assume we can capture shape by boundary and rough shading



Then, just add another constraint when sampling: similarity to underlying image at that spot

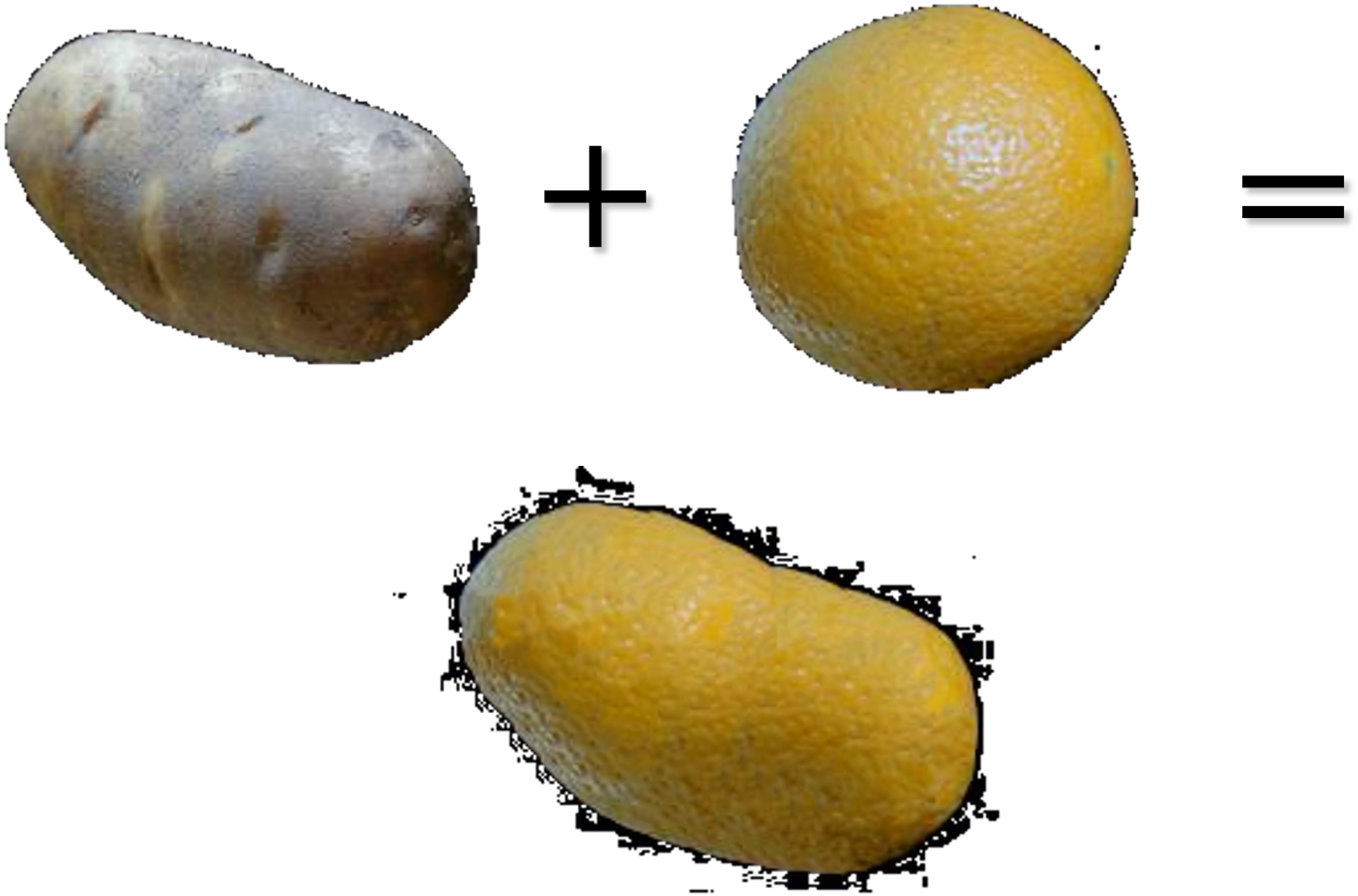
Slide credit A. Efros

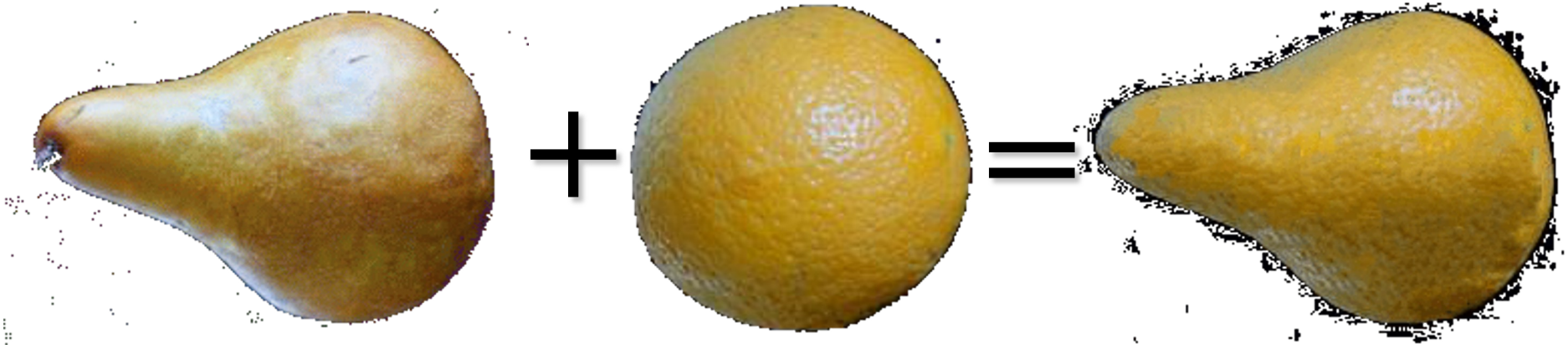
parmesan

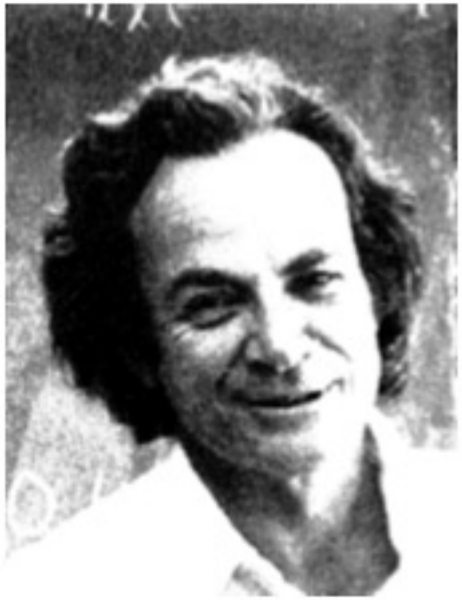


rice

Slide credit A. Efros





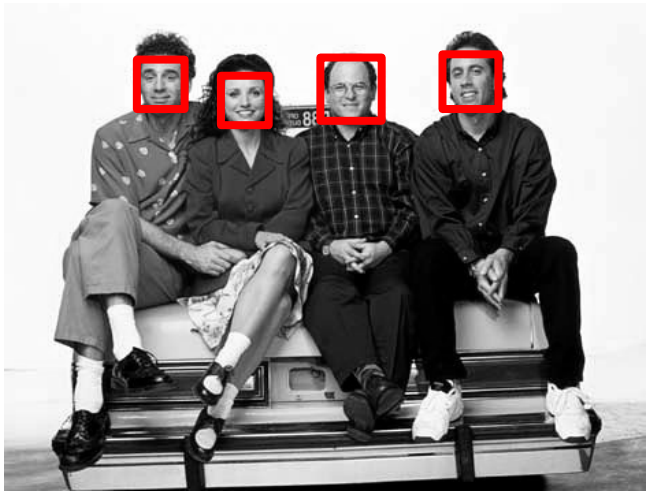


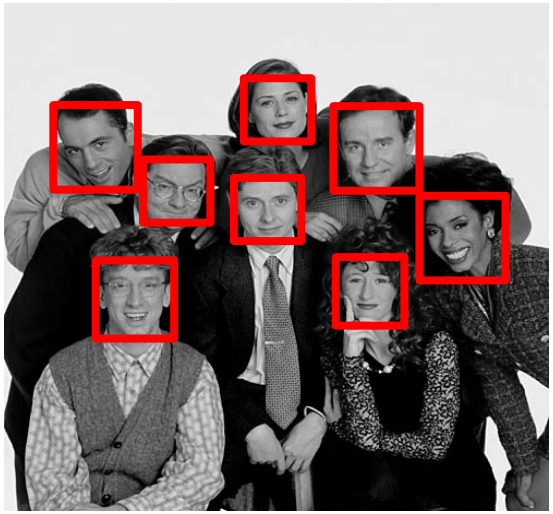
3. Face detection using sliding window and histograms of oriented gradients

Where are the faces in an image?

Face detection using sliding window and histograms of oriented gradients

Where are the faces in an image?

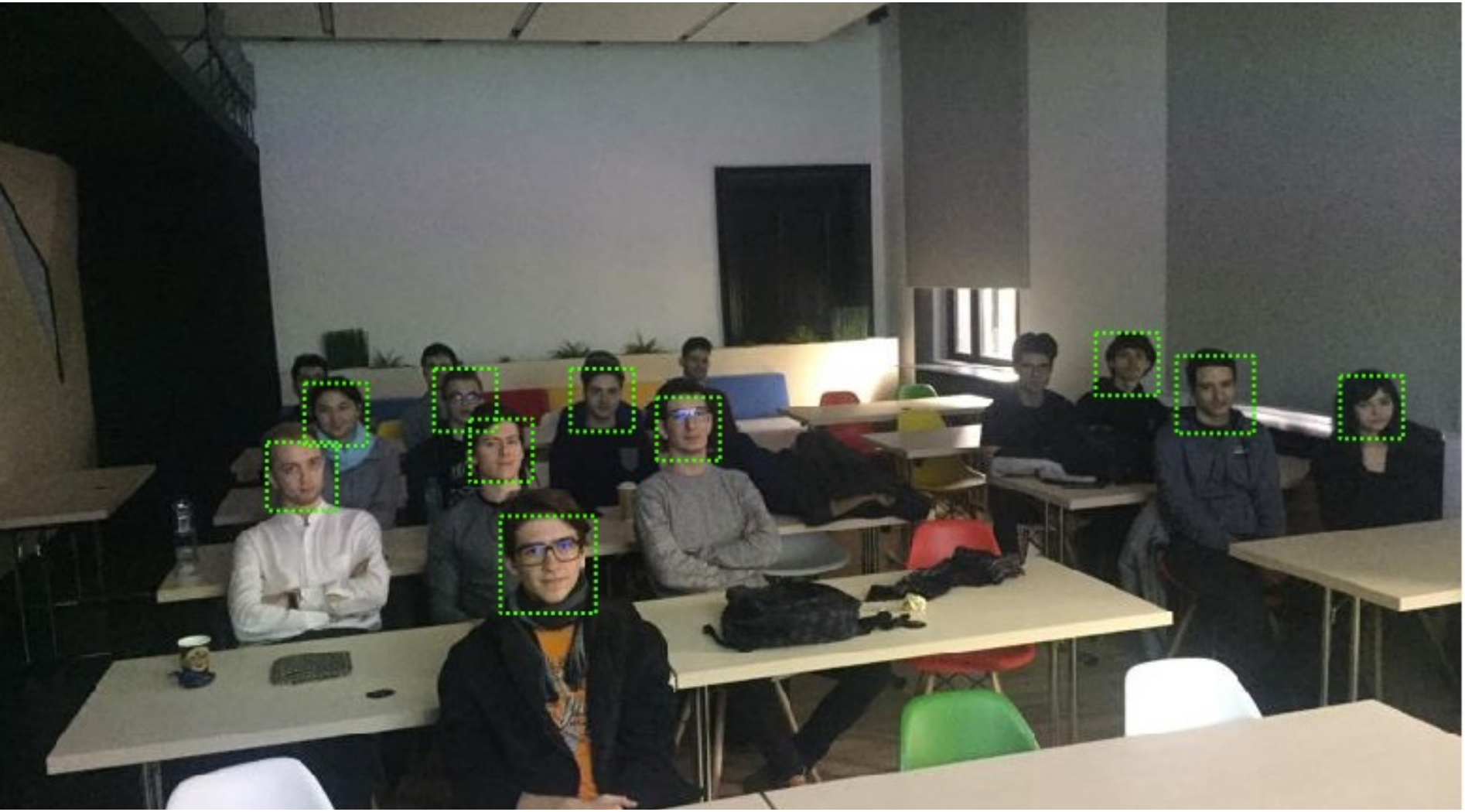




Localization done at the level of a bounding box.

Object detection using sliding window

- detection via classification: binary classifier for face → does a window contain a face?
- consider windows positioned at each pixel, for different sizes (to achieve scale invariance)

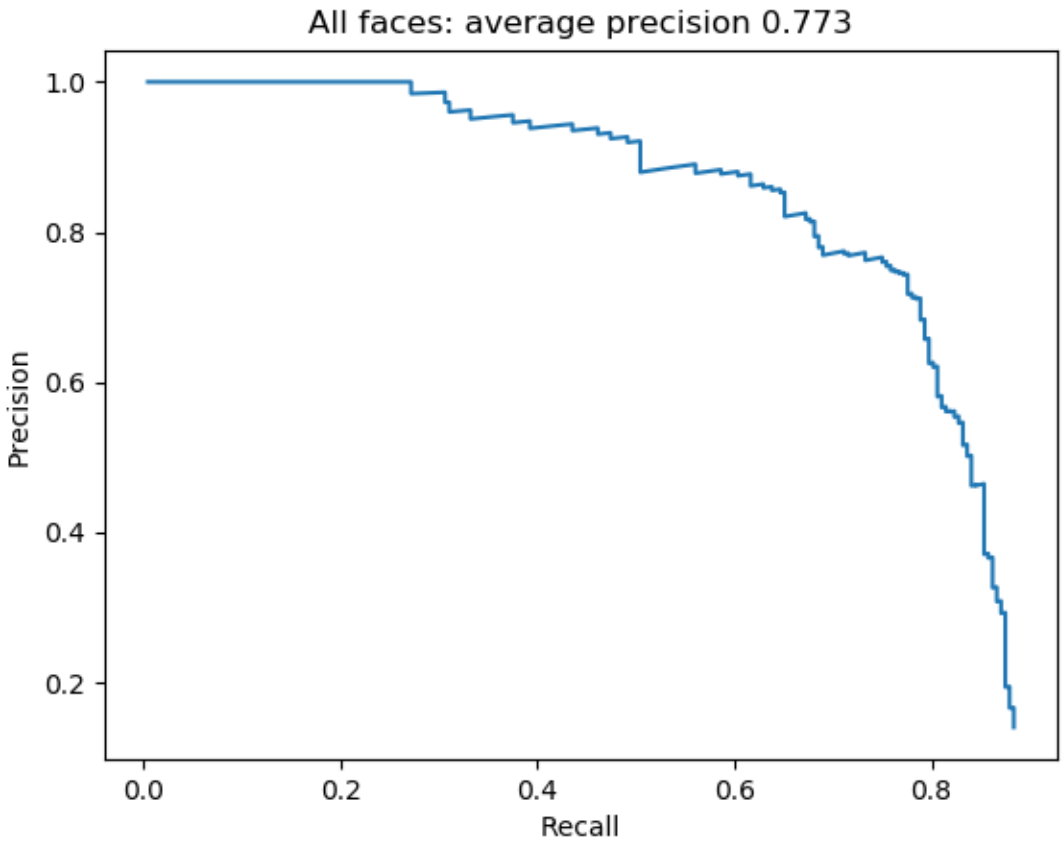


4. Face detection and recognition in cartoons

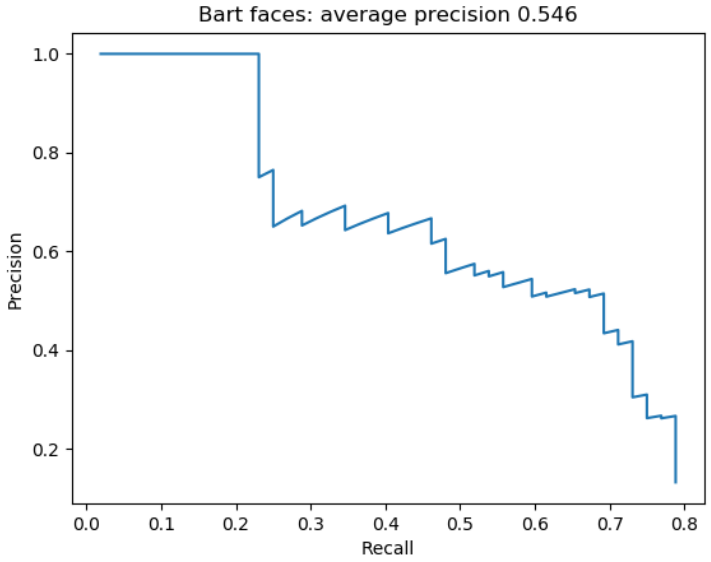
face detection = detect all faces

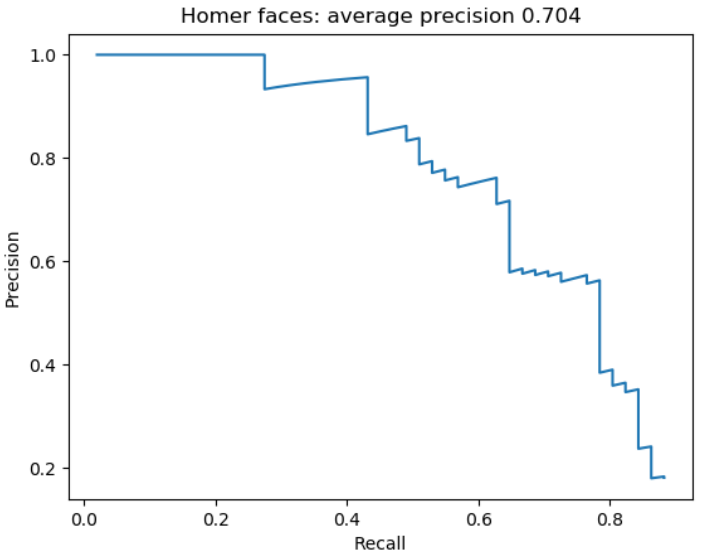
face recognition = detect specific faces (Bart, Lisa, Marge, Homer)

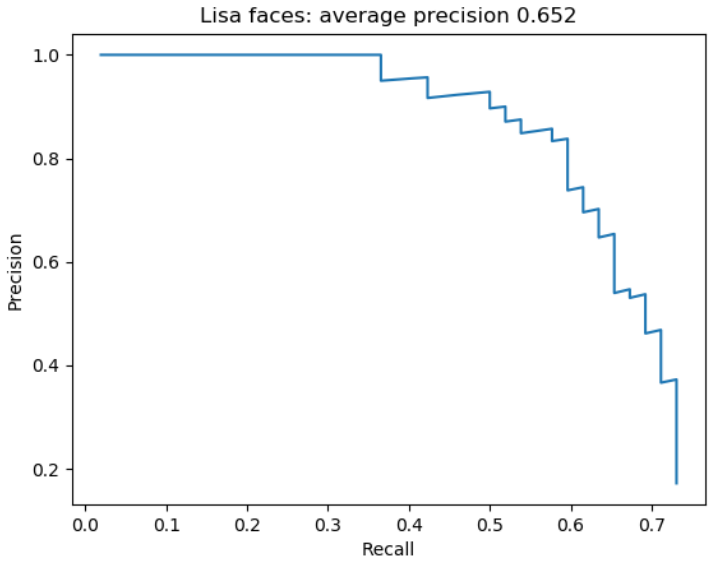
Average precision

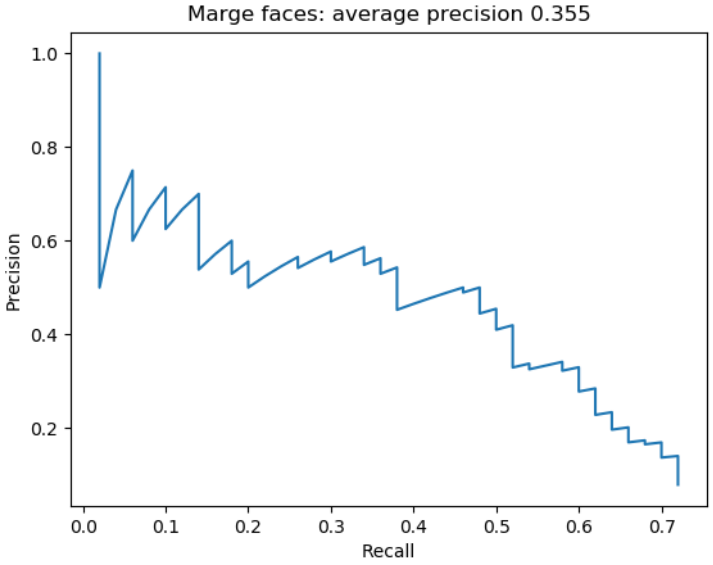


Mean average precision









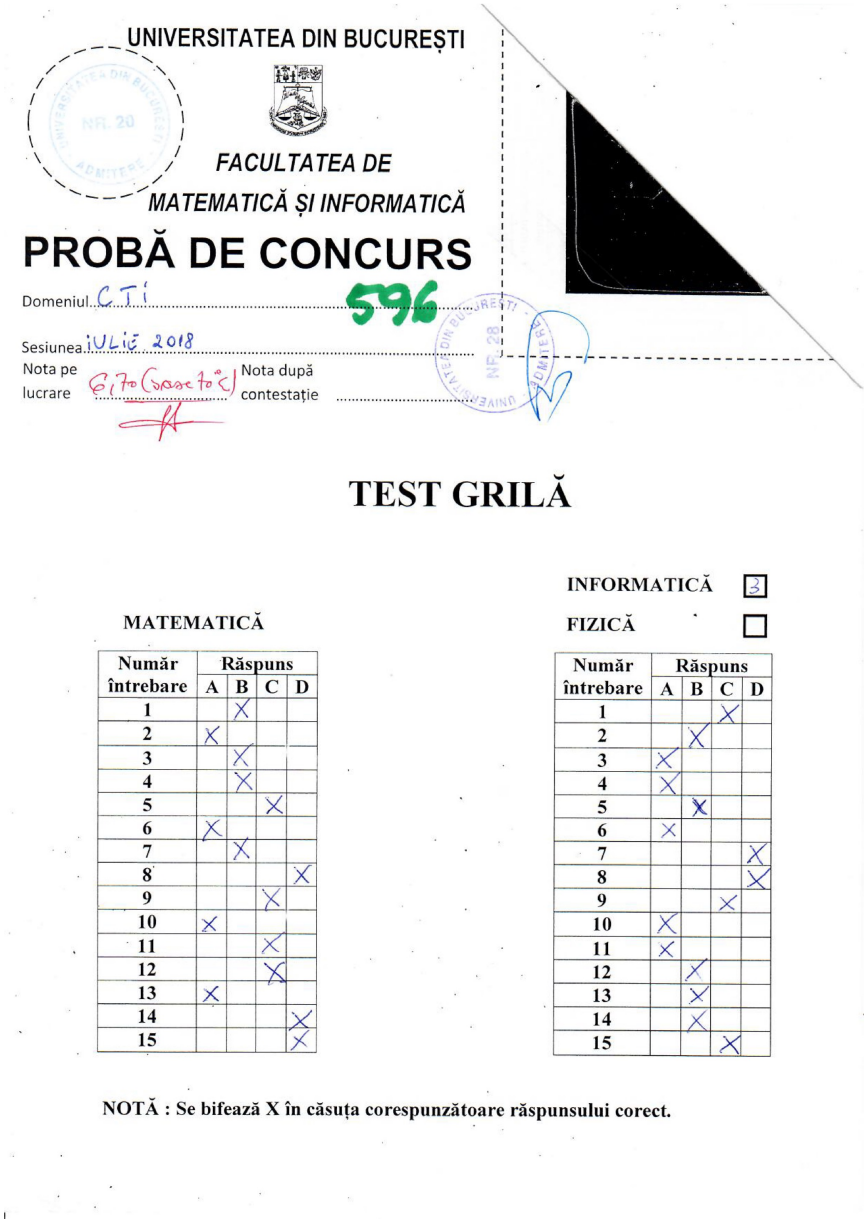
mAP = take the mean of the four APs

Computer Vision projects for master students

Computer Vision projects for master students: Project 1 and 2 proposed last years

2019-2020	https://tinyurl.com/CV-2020-Project1 https://tinyurl.com/CV-2020-Project2
2020-2021	https://tinyurl.com/CV-2021-Project1 https://tinyurl.com/CV-2021-Project2
2021-2022	https://tinyurl.com/CV-2022-Project1 https://tinyurl.com/CV-2022-Project2
2022-2023	https://tinyurl.com/CV-2023-Project1 https://tinyurl.com/CV-2023-Project2

1. Automatic grading of multiple choice tests



UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

PROBĂ DE CONCURS

Domeniul CTI

596

Sesiunea IULIE 2018

Nota pe lucrare 6,7 (șase virgulă șapte)

Nota după contestație

TEST GRILĂ

INFORMATICĂ [3]

FIZICĂ [ ]

MATEMATICĂ

Număr întrebare	Răspuns A	B	C	D
1		X		
2	X			
3		X		
4		X		
5			X	
6	X			
7		X		
8				X
9			X	
10	X			
11			X	
12			X	
13	X			
14				X
15				X

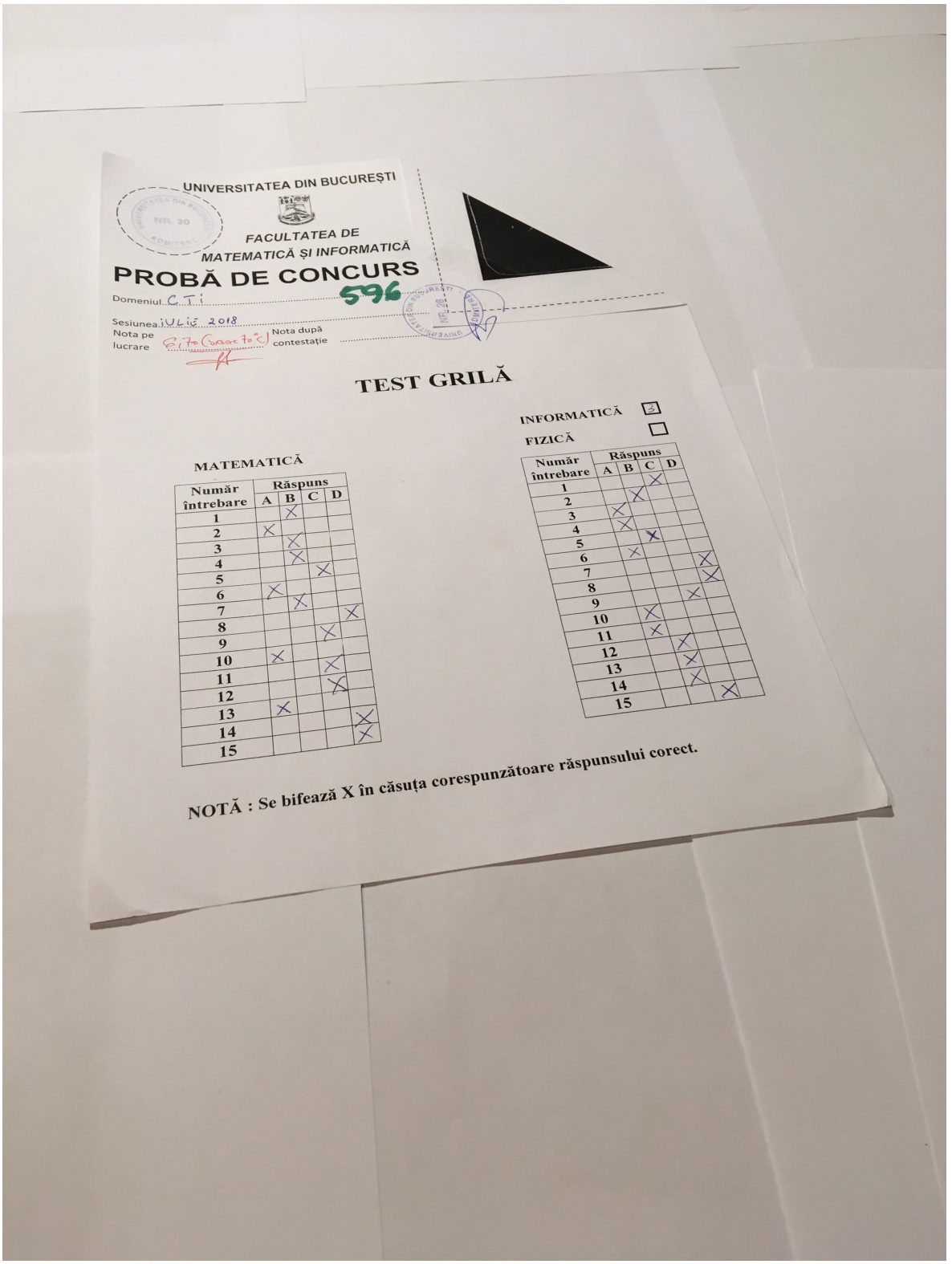
Număr întrebare	Răspuns A	B	C	D
1			X	
2		X		
3	X			
4	X			
5		X		
6	X			
7				X
8				X
9			X	
10	X			
11	X			
12		X		
13		X		
14		X		
15			X	

NOTĂ : Se bifează X în căsuța corespunzătoare răspunsului corect.

image_2.txt

```
I 3
1 B
2 A
3 B
4 B
5 C
6 A
7 B
8 D
9 C
10 A
11 C
12 C
13 A
14 D
15 D
16 C
17 B
18 A
19 A
20 B
21 A
22 D
23 D
24 C
25 A
26 A
27 B
28 B
29 B
30 C
R 19
```

Rotation + scaling

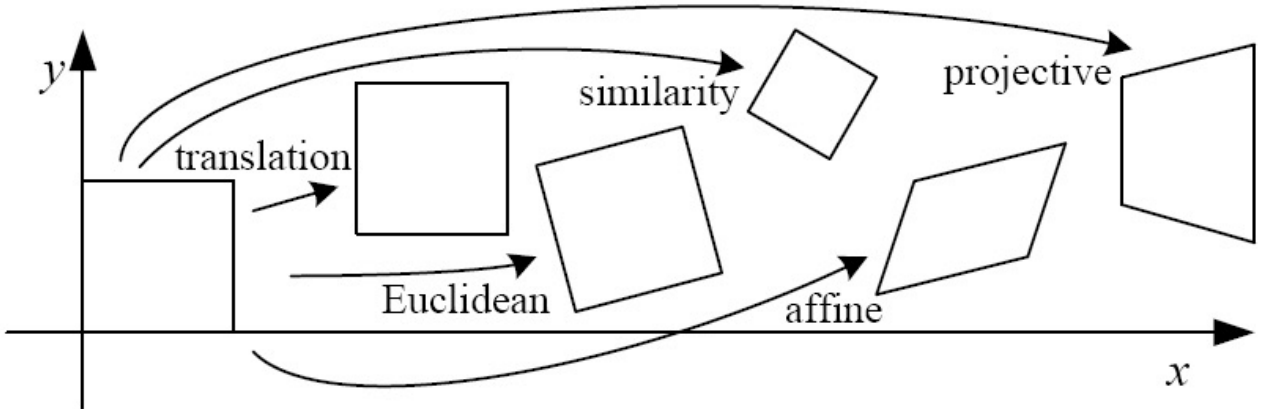


Număr întrebare	Răspuns A	B	C	D
1		X		
2	X			
3		X		
4		X		
5			X	
6	X			
7		X		
8				X
9			X	
10	X			
11			X	
12			X	
13	X			
14				X
15				X

Număr întrebare	Răspuns A	B	C	D
1			X	
2		X		
3	X			
4	X			
5			X	
6		X		
7				X
8				X
9			X	
10		X		
11		X		
12			X	
13			X	
14			X	
15				X

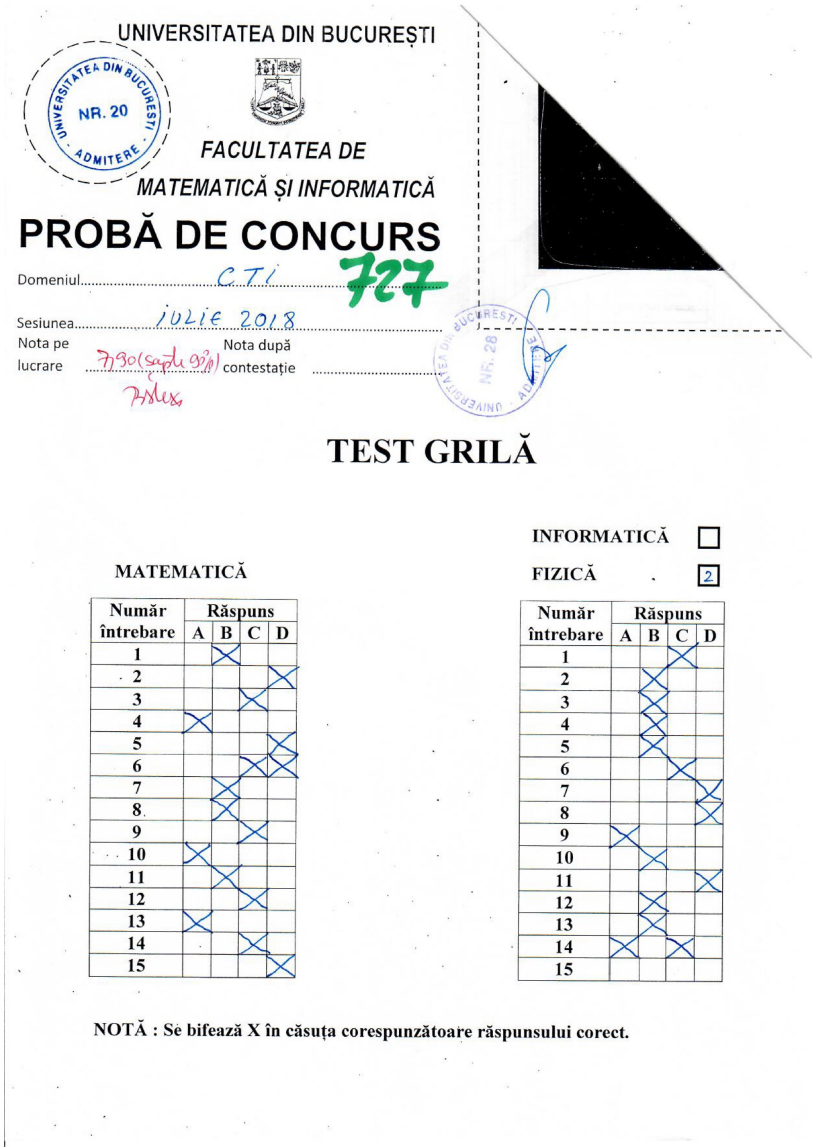
Perspective

2D Geometric transformations



Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\left[\begin{array}{c\|c} \boldsymbol{I} & \boldsymbol{t} \end{array}\right]_{2\times 3}$	2	orientation $+\cdots$	
rigid (Euclidean)	$\left[\begin{array}{c\|c} \boldsymbol{R} & \boldsymbol{t} \end{array}\right]_{2\times 3}$	3	lengths $+\cdots$	
similarity	$\left[\begin{array}{c\|c} s\boldsymbol{R} & \boldsymbol{t} \end{array}\right]_{2\times 3}$	4	angles $+\cdots$	
affine	$\left[\begin{array}{c} \boldsymbol{A} \end{array}\right]_{2\times 3}$	6	parallelism $+\cdots$	
projective	$\left[\begin{array}{c} \tilde{\boldsymbol{H}} \end{array}\right]_{3\times 3}$	8	straight lines	

BONUS: handwritten grade recognition



UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

PROBĂ DE CONCURS

Domeniul CTI 727

Sesiunea IULIE 2018

Nota pe lucrare 7,90 (șapte 9/10) Nota după contestație

TEST GRILĂ

INFORMATICĂ ☐

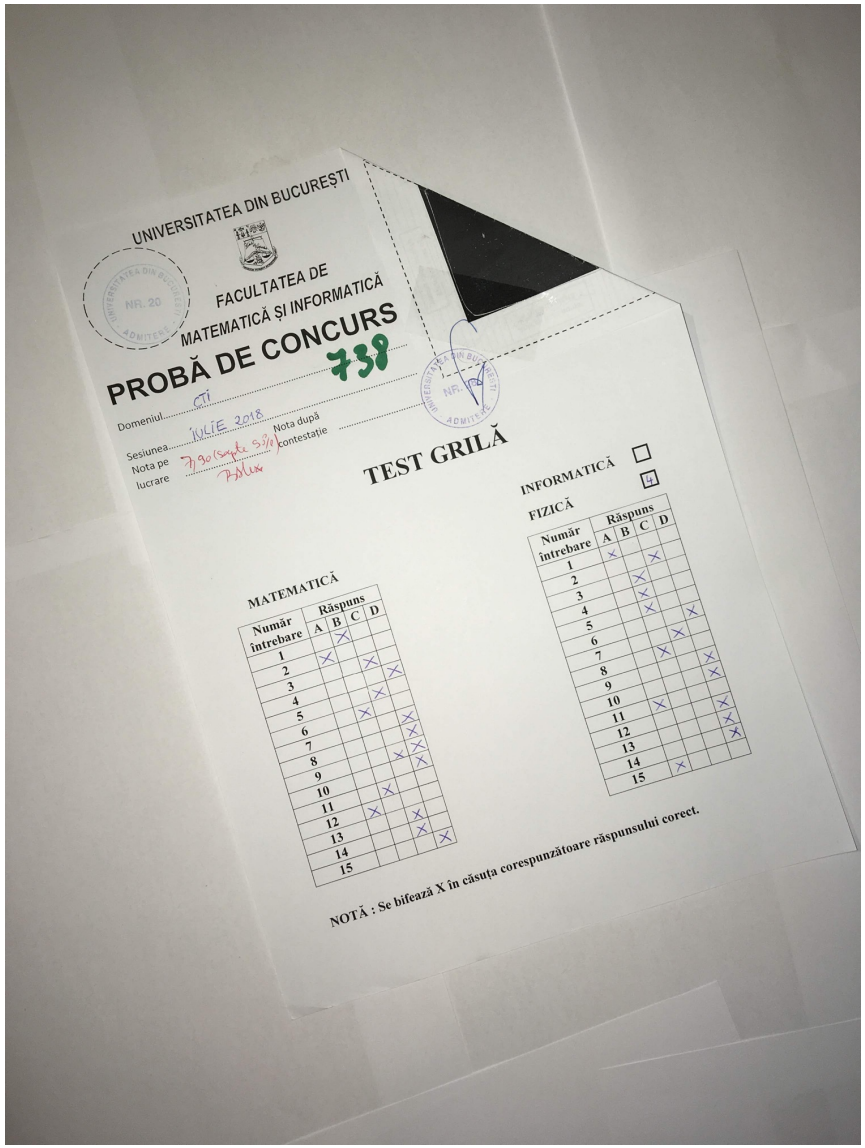
FIZICĂ 2

MATEMATICĂ

Număr întrebare	Răspuns A	B	C	D
1		X		
2				X
3			X	
4	X			
5				X
6			X	X
7		X		
8		X		
9			X	
10	X			
11		X		
12			X	
13	X			
14			X	
15				X

Număr întrebare	Răspuns A	B	C	D
1			X	
2		X		
3		X		
4		X		
5		X		
6			X	
7				X
8				X
9	X			
10		X		
11				X
12		X		
13		X		
14	X		X	
15				

NOTĂ : Se bifează X în căsuța corespunzătoare răspunsului corect.



UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

PROBĂ DE CONCURS 738

Domeniul CTI

Sesiunea IULIE 2018

Nota pe lucrare 7,30 (șapte 3/10) Nota după contestație

TEST GRILĂ

INFORMATICĂ ☐

FIZICĂ 4

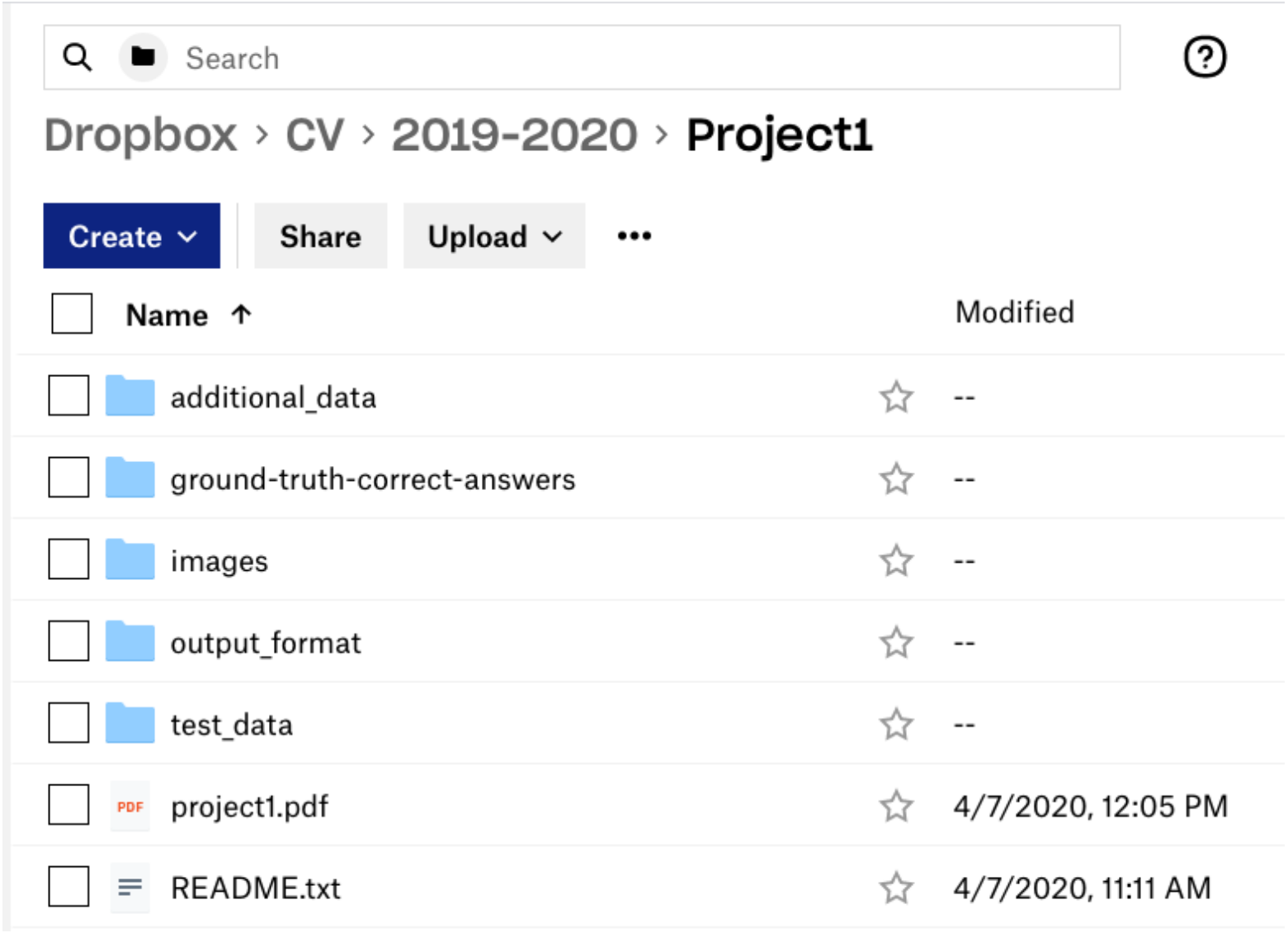
MATEMATICĂ

Număr întrebare	Răspuns A	B	C	D
1	X			
2				X
3			X	
4			X	
5	X			
6				X
7				X
8				X
9				
10		X		
11	X		X	
12				X
13			X	
14		X		
15				

Număr întrebare	Răspuns A	B	C	D
1		X		
2			X	
3			X	
4			X	
5				X
6			X	
7				X
8				X
9				
10	X			X
11				X
12				X
13				
14	X			
15				

NOTĂ : Se bifează X în căsuța corespunzătoare răspunsului corect.

Project 1: tinyurl.com/CV-2020-project1



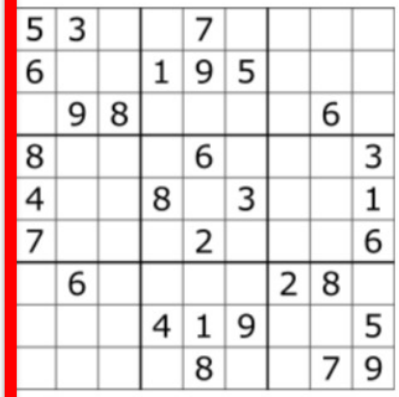
2. Extracting visual information from Sudoku puzzles - tinyurl.com/CV-2021-Project1

Sudoku is currently one of the most famous puzzles in the world. The most popular version consists on a 9×9 grid made up of 3×3 subgrids, but the general case, an $n^2 \times n^2$ grid with $n \times n$ subgrids is considered. Some cells contain numbers, which can be considered as input data. The goal is to fill in the empty cells, one number in each, so that each column, row, and subgrid contains the numbers 1 through 9 exactly once (numbers 1 to n^2 in the general case). If the input data are correct, the sudoku has one and only one solution.

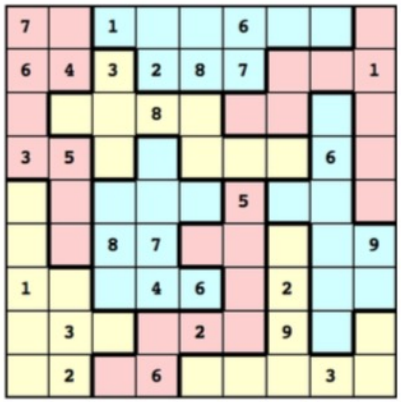
5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

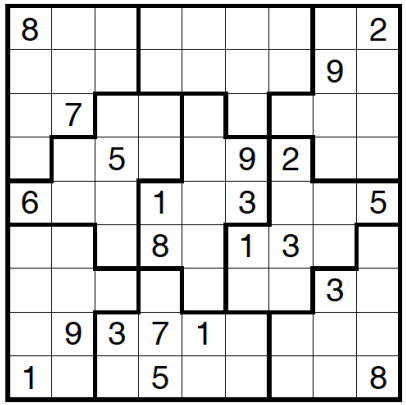
Many variants of Sudoku puzzles



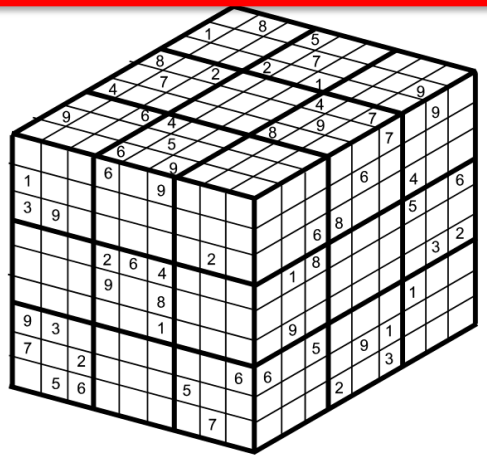
classic



jigsaw

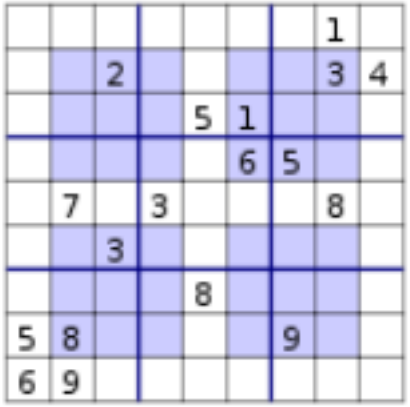


jigsaw

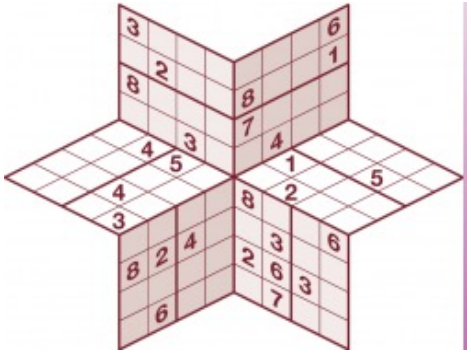


cube

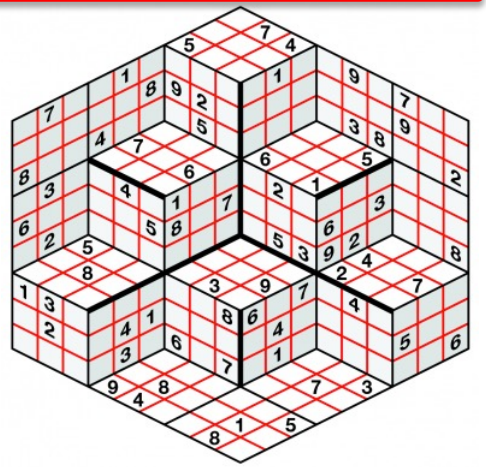
wordoku



hypersudoku



3D star



3D

- many more…

First task - Classic Sudoku

In the first task you are asked to write a program that processes an input image containing a Classic Sudoku puzzle and outputs the configuration of the puzzle by marking the empty cells with letter 'o' and the filled in cells with letter 'x'. The training data consists of 50 training examples. Each training example (an image obtained by taking a photo with the mobile phone) contains one Classic Sudoku puzzle, centered, usually axis aligned or with small rotations with respect to the Ox and Oy axis. Figure 4 shows two training examples of Clasic Sudoku puzzles and their corresponding configurations.

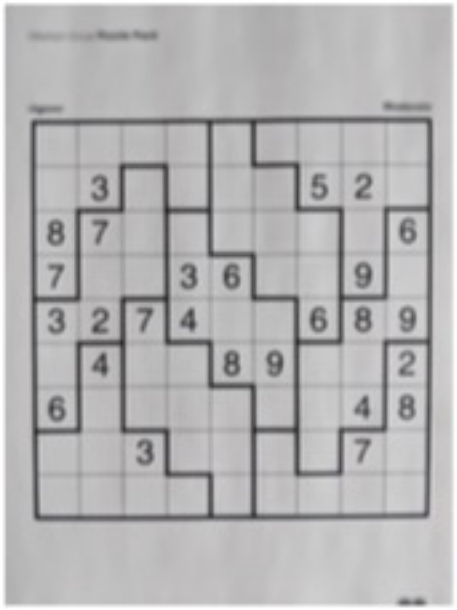
```
OOXXOXXOO
XOXOOOOXX
OOOOXOXXO
XOXOXOOXO
XOOOXOOOX
OXOOXOXOX
OXXOXOOOO
XXOOOOXOX
OOXXOXXOO
```

```
OXOOXOOXO
XOOXOXOOX
OOXOOOXOO
OXOOXOOXO
XOXXOXXOX
OXOOXOOXO
OOXOOOXOO
XOOXOXOOX
OXOOXOOXO
```

Second task - Jigsaw Sudoku

In the second task you are asked to write a program that processes an input image containing a Jigsaw Sudoku puzzle and outputs the configuration of the puzzle by: (1) determining the irregular shape regions in the puzzle and marking them with digits from 1 to 9; (2) marking the empty cells with letter 'o' and the filled in cells with letter 'x'. The irregular shape regions from the puzzle are separated by bold borders and sometimes (in the colored puzzles) contain cells with the same color.

```
1o1x1o1o1x1o1o2o2o
3o3o1o4x4x1o4x4o2o
3o3x3o3x4o4o4x2x2x
5o5o3x3o3o6o4o4o2o
5x5o5x6o6x6x2o2x2o
5o5o6x6o6o7o7o7o8o
9o5o6o7o7o7o7o8o8o
9x5o6x7o9o9o7o8x8x
9x9o9o9o9o8o8o8o8o
```

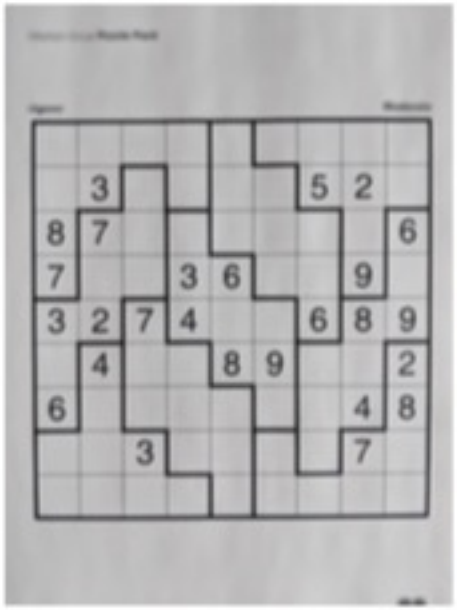


```
1o1o1o1o2o3o3o3o3o
1o1x4o1o2o2o3x3x3o
1x4x4o5o2o2o2o3o6x
1x4o4o5x5x2o2o3x6o
4x4x7x5xmo5o2x6x6x
4o8x7o7o5x5x6o6o9x
4x8o7o7o7o5o6o6x9x
8o8o8x7o7o9o6o9x9o
8o8o8o8o7o9o9o9o9o
```

Second task - Jigsaw Sudoku

red puzzles) contain cells with the same color. For marking the irregular shape regions in a Jigsaw puzzle we use the following simple algorithm: (i) we process the cells from left to right and top to bottom; (ii) the top left cell gets digit 1 as it is part of region 1; (iii) we assign the same digit for all cells in the same region; (iv) the first cell in the next region gets the increased digit.

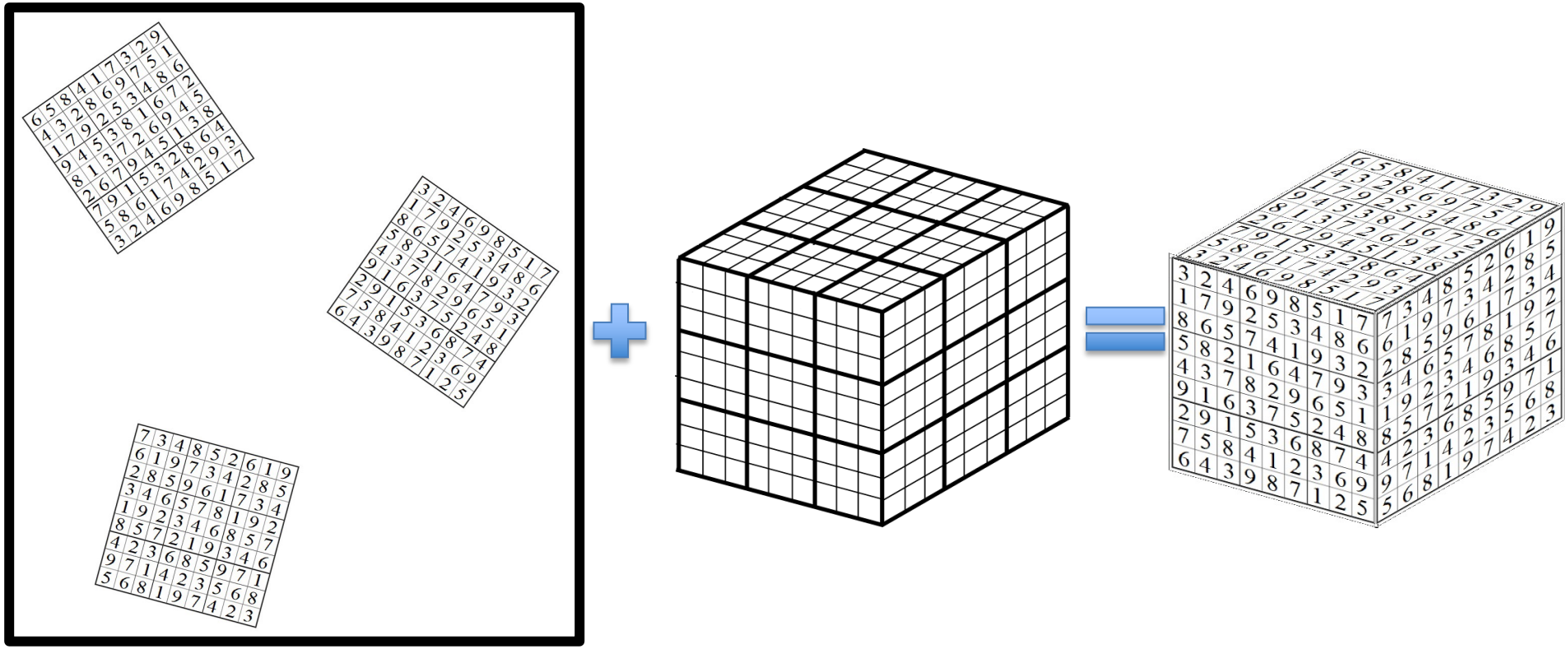
```
1o1x1o1o1x1o1o2o2o
3o3o1o4x4x1o4x4o2o
3o3x3o3x4o4o4x2x2x
5o5o3x3o3o6o4o4o2o
5x5o5x6o6x6x2o2x2o
5o5o6x6o6o7o7o7o8o
9o5o6o7o7o7o7o8o8o
9x5o6x7o9o9o7o8x8x
9x9o9o9o9o8o8o8o8o
```



```
1o1o1o1o2o3o3o3o3o
1o1x4o1o2o2o3x3x3o
1x4x4o5o2o2o2o3o6x
1x4o4o5x5x2o2o3x6o
4x4x7x5xmo5o2x6x6x
4o8x7o7o5x5x6o6o9x
4x8o7o7o7o5o6o6x9x
8o8o8x7o7o9o6o9x9o
8o8o8o8o7o9o9o9o9o
```

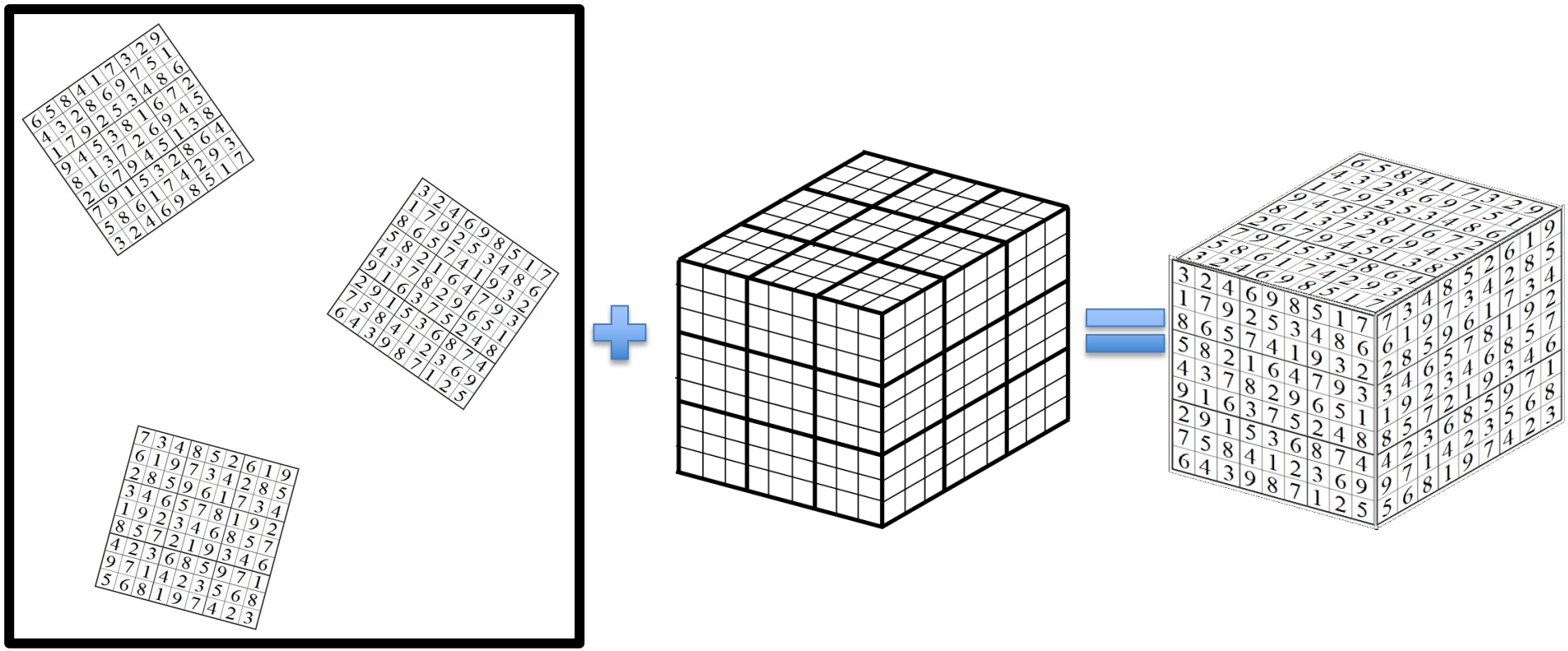
Third task – Sudoku Cube

In the third task you are asked to write a program that processes an input image containing three sides (sudoku puzzles) of a Sudoku Cube and outputs the coresponding Sudoku Cube by: (1) localizing the three sudoku puzzles in the image that form the sides of the Sudoku Cube; (2) inferring their position in the Sudoku Cube using the constraint that the digits on the common edge of two sides must be the same number; (3) warping each side on the corresponding side of a given template for the Sudoku Cube. The training data



Third task – Sudoku Cube

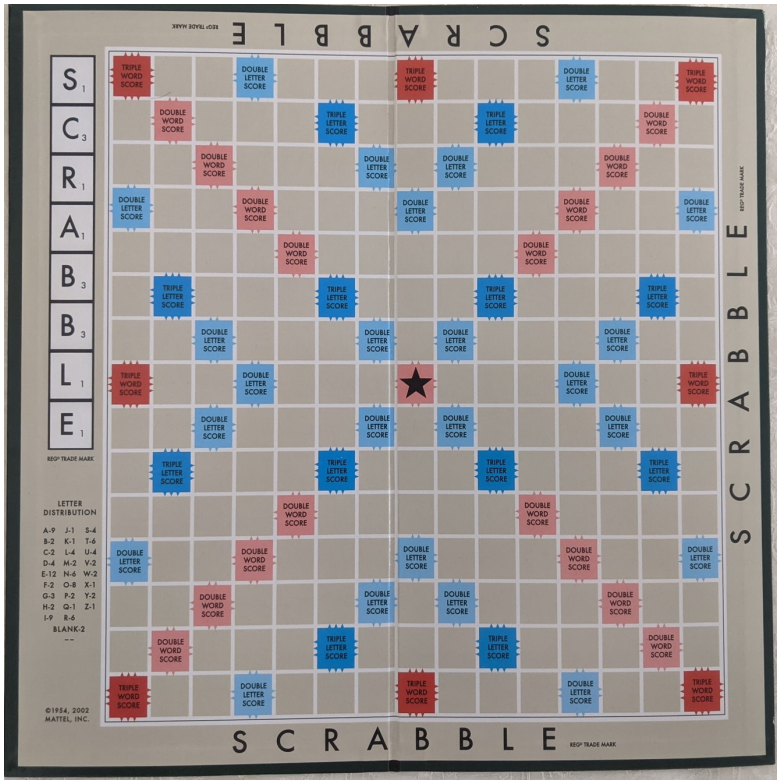
on the corresponding side of a given template for the Sudoku Cube. The training data consists of 10 training examples. Each training example (an image obtained by taking a photo with the mobile phone) contains three sides of a Sudoku Cube. They are scattered around the image and usually rotated with respect to the axis. Figure 6 shows the Sudoku Cube template, the input image containing the three sides and the desired output. You are allowed to annotate points on the template for warping.



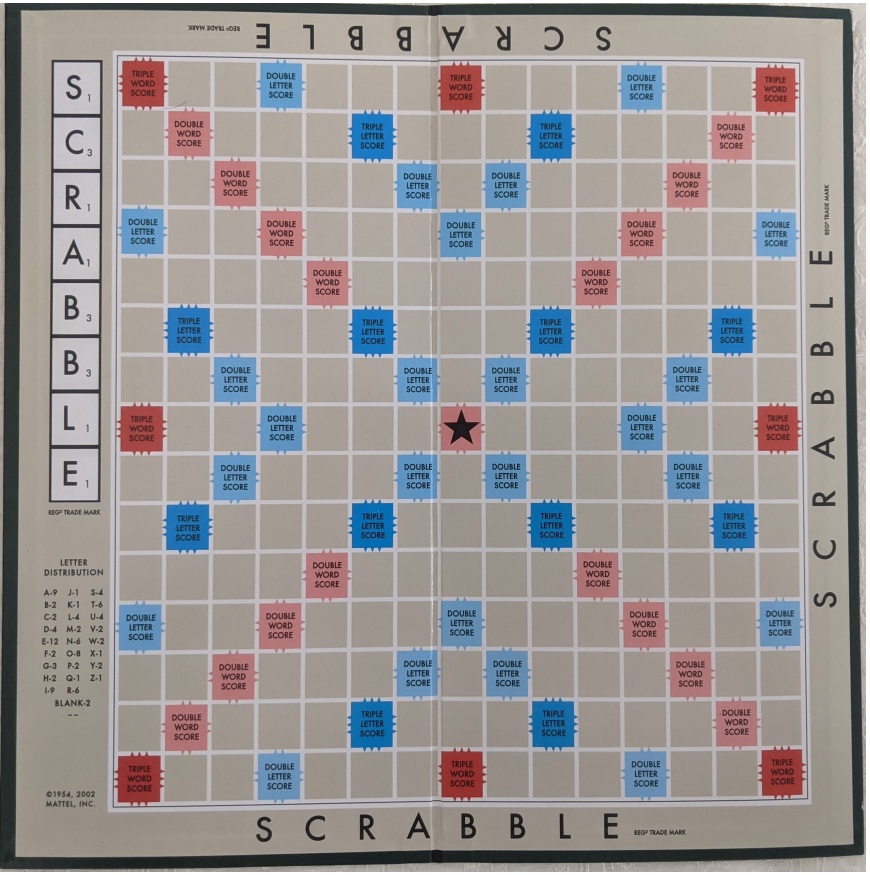
3. Scrabble word score calculator

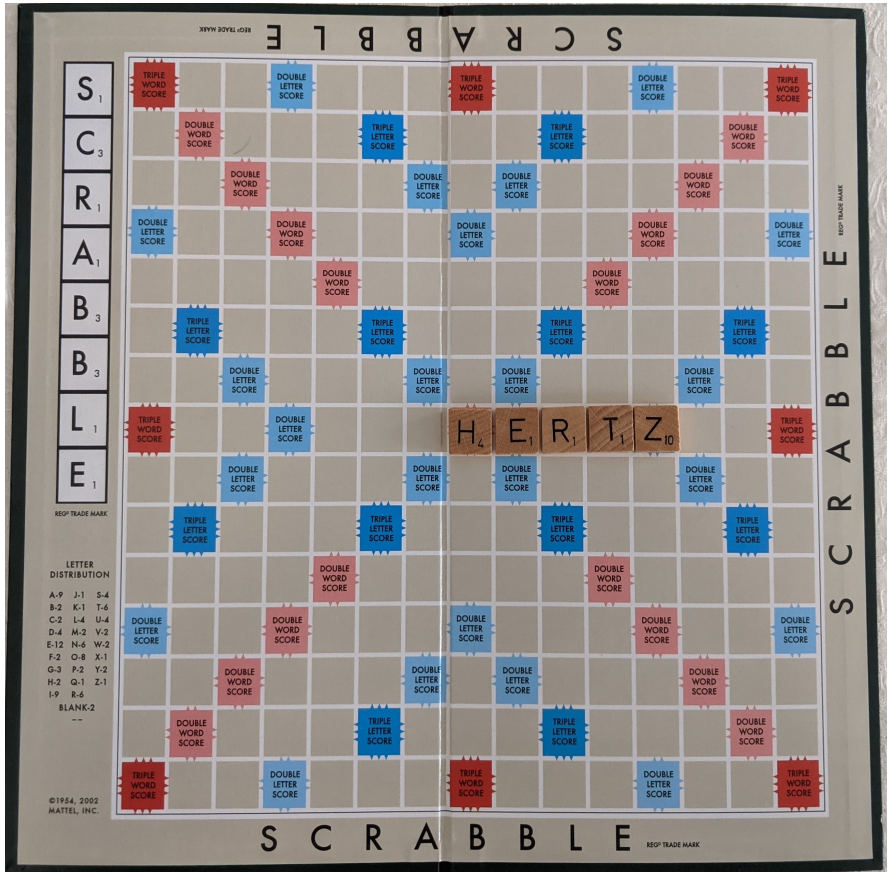
https://tinyurl.com/CV-2022-Project1

Scrabble is a word game for two, three of four players. The play consists of forming interlocking words, cross-word fashion, on the Scrabble playing board (Figure 1) using letter tiles with various score values. The tiles must form words that, in crossword fashion, read left to right in rows or downward in columns, and be included in a standard dictionary or lexicon. Each player competes for high score by using his letters in combinations and locations that take best advantage of letter values and premium squares on the board.



Examples of words formation and scoring





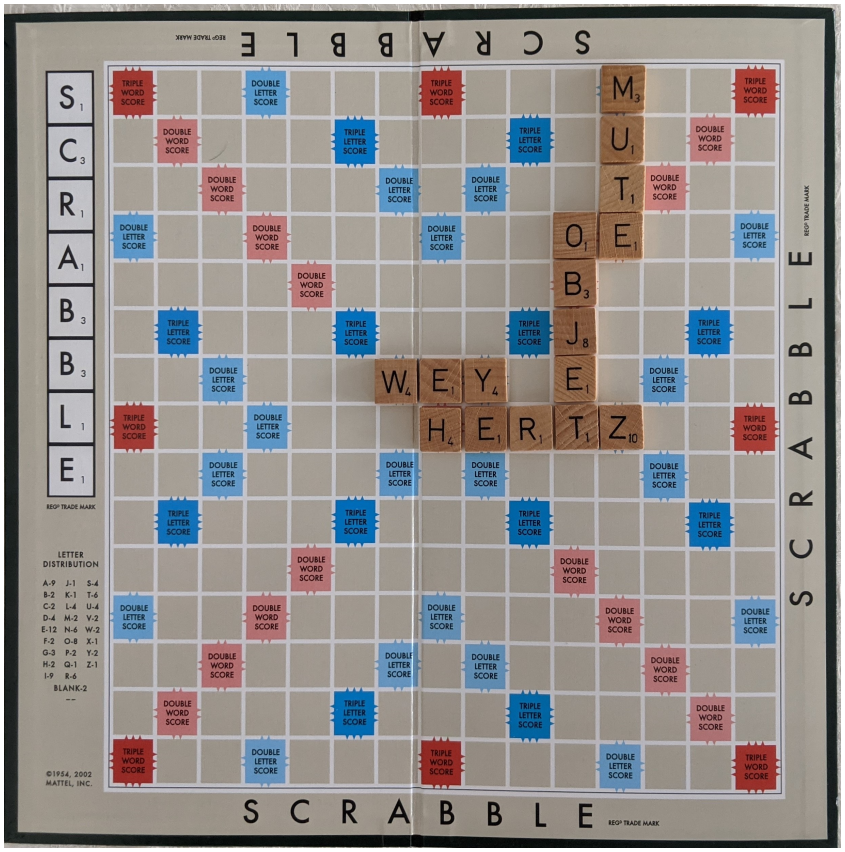
HERTZ: (4(H) + 1(E) + 1(R) + 1(T) + 10(Z)*2(DLS))*2(DWS) = 54

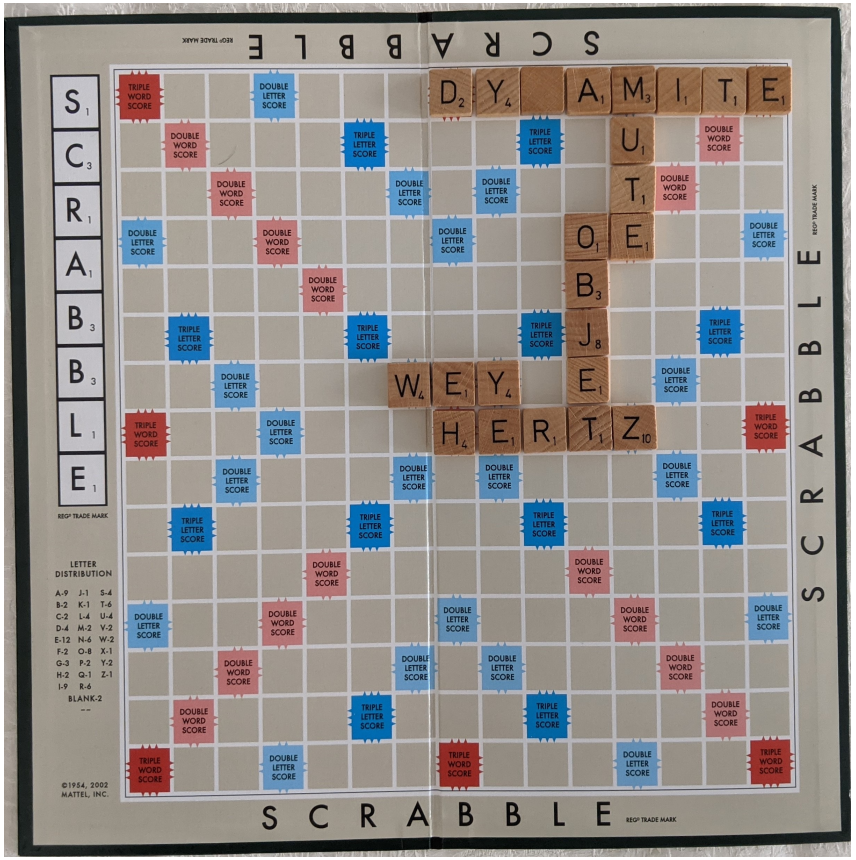
TOTAL: 54

DLS = DOUBLE LETTER SCORE
TLS = TRIPLE LETTER SCORE

DWS = DOUBLE WORD SCORE
TWS = TRIPLE WORD SCORE

Examples of words formation and scoring





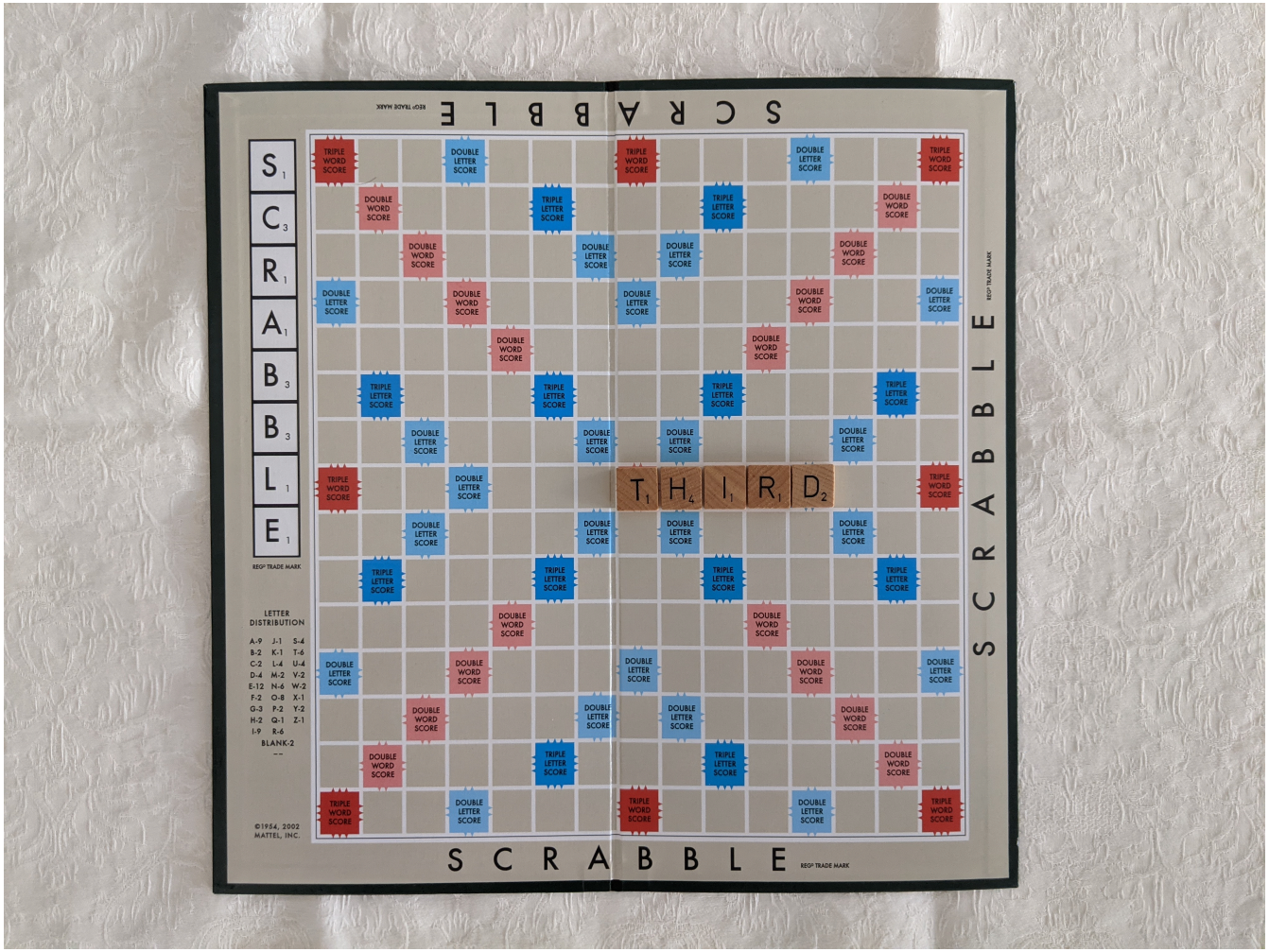
DYnAMITE: (2(D) + 4(Y) + 0(n) + 1(A) + 3(M) + 1(I) + 1(T) + 1(E)) *3(TWS) * 3(TWS) + 50 (bonus) = 13*9 + 50 = 167

TOTAL: 167

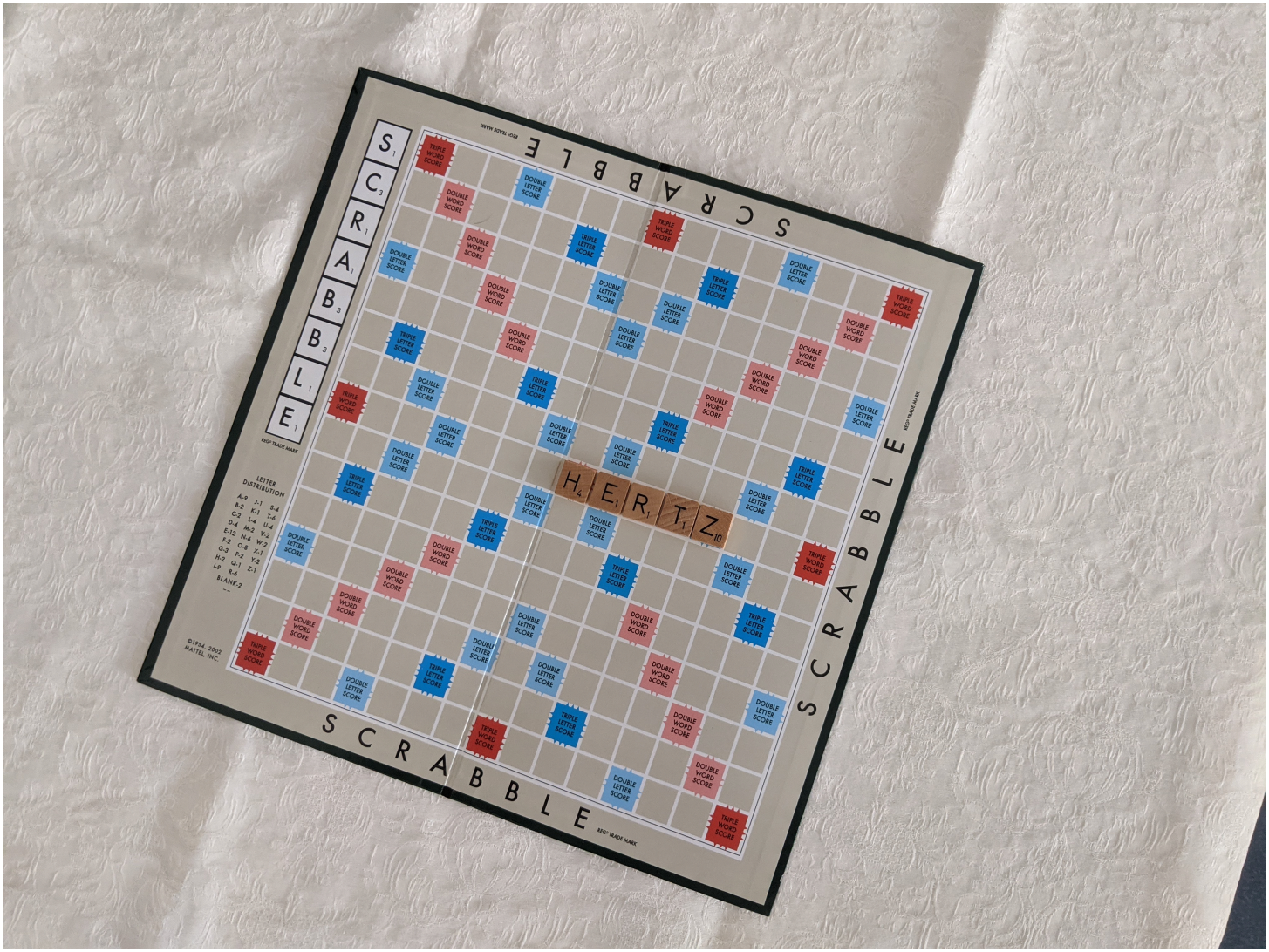
DLS = DOUBLE LETTER SCORE
TLS = TRIPLE LETTER SCORE

DWS = DOUBLE WORD SCORE
TWS = TRIPLE WORD SCORE

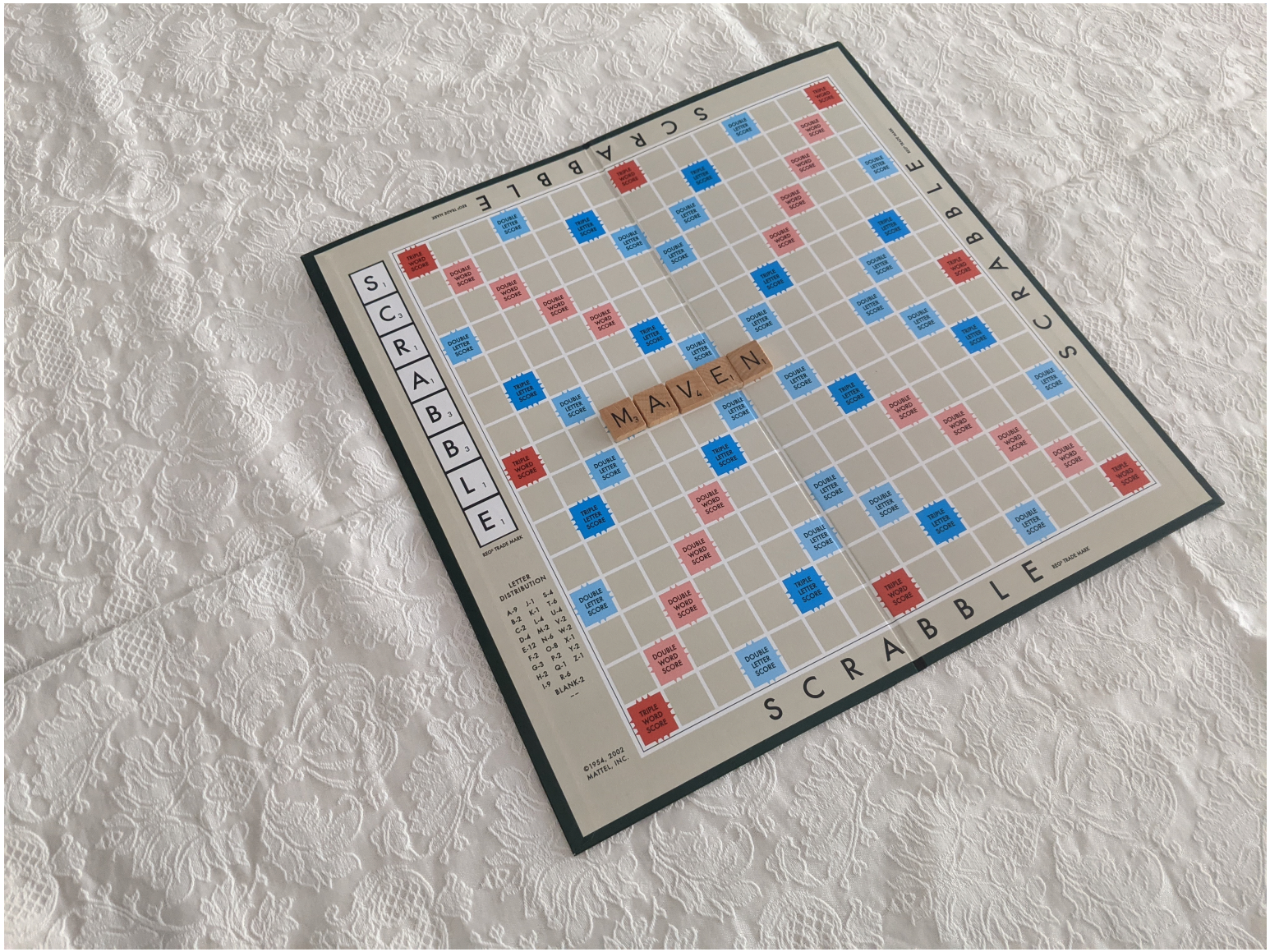
Regular view



Rotated view



Perspective view

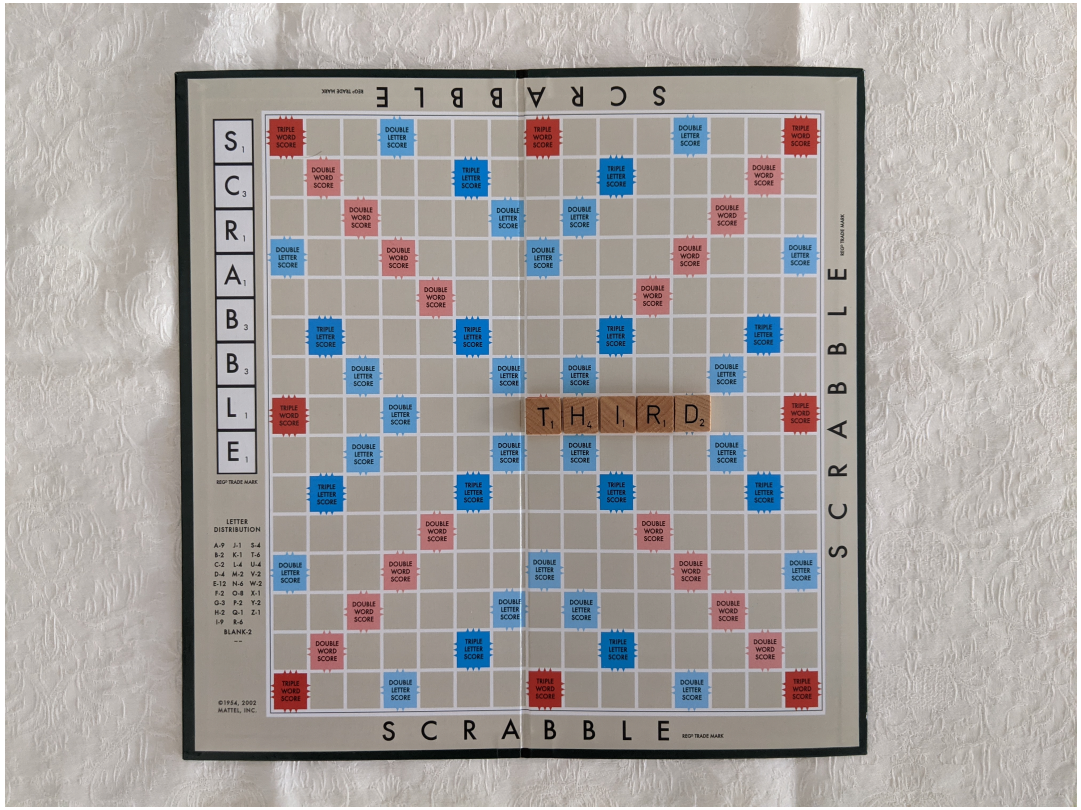


Ground-truth annotations

For each image we annotate:

1. the positions of the tiles added to the board (form left to right, up to down) – use notations 1-15 for rows and A-O for columns
2. the tile letter
3. the total score

1_01.jpg



1_01.txt

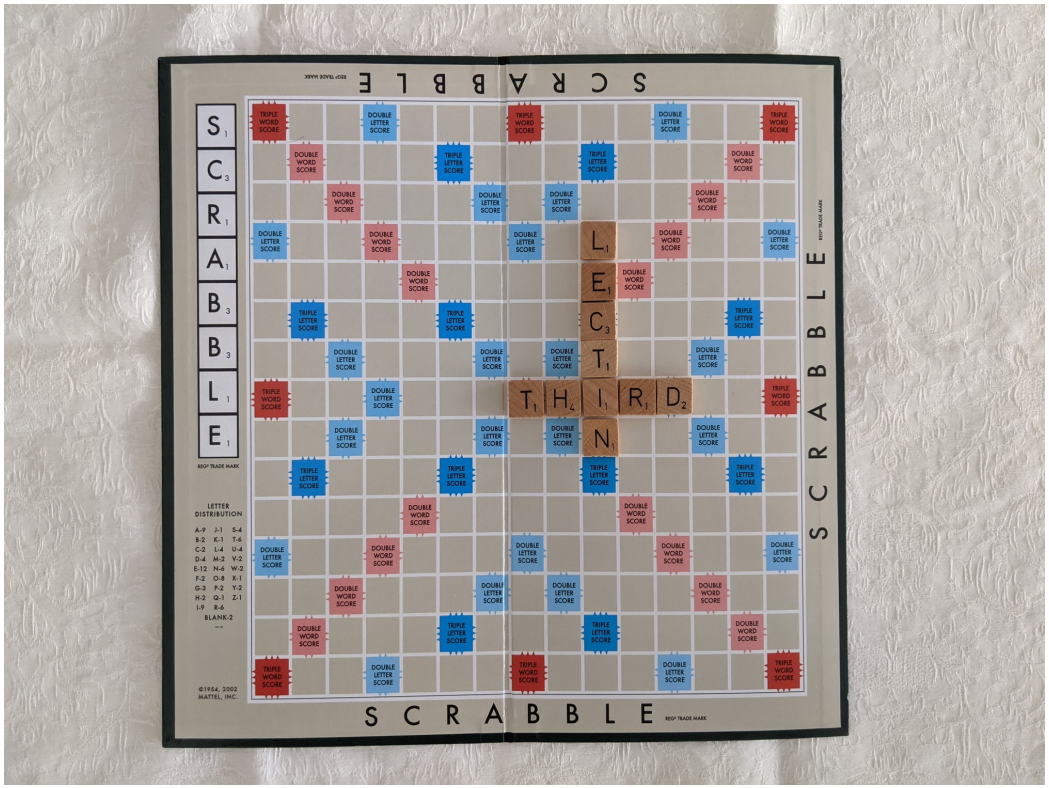
```
8H T
8I H
8J I
8K R
8L D
22
```

Ground-truth annotations

For each image we annotate:

1. the positions of the tiles added to the board (form left to right, up to down) – use notations 1-15 for rows and A-O for columns
2. the tile letter
3. the total score

1_02.jpg



1_02.txt

```
4J  L
5J  E
6J  C
7J  T
9J  N
14
```

Test Data

-same as training data: 5 games, each game has 20 turns (1 turn = 1 image)
-testing data: 100 images (will be made available after the deadline for submitting the source code, on Friday 29th of April)

For each test image you should produce a similar txt file (line in the annotations) containing:

1. the positions of the tiles added to the board (form left to right, up to down) – use notations 1-15 for rows and A-O for columns
2. the tile letter
3. the total score

Each correctly solved image worth 0.05 points:

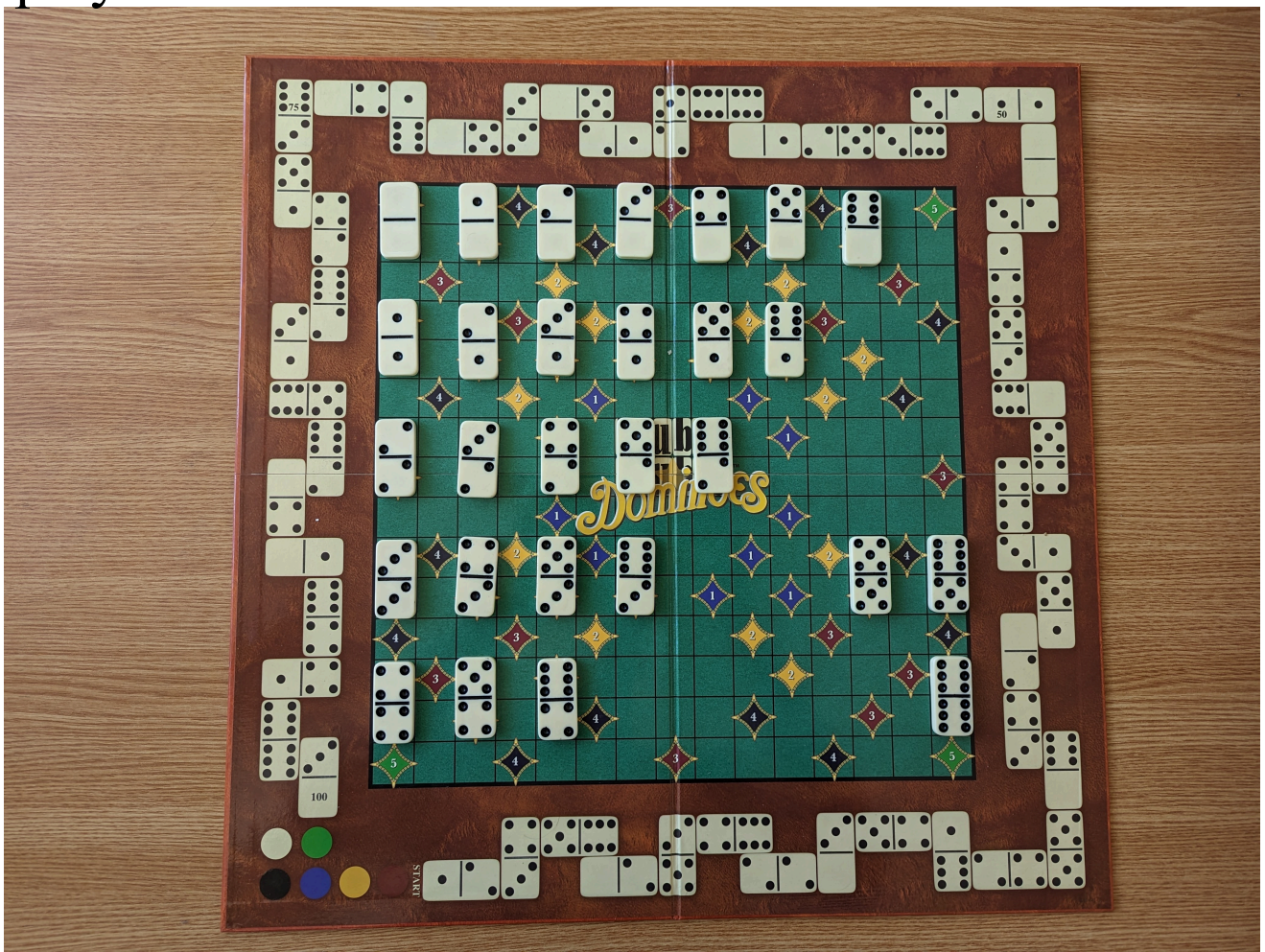
- correct ALL positions: 0.025 points
- correct ALL letters: 0.015 points
- correct total score: 0.01 points

Total: 100 x 0.05 = 5 points
Ex officio: 0.5 points (correct format)

4.Double Double Dominoes score calculator

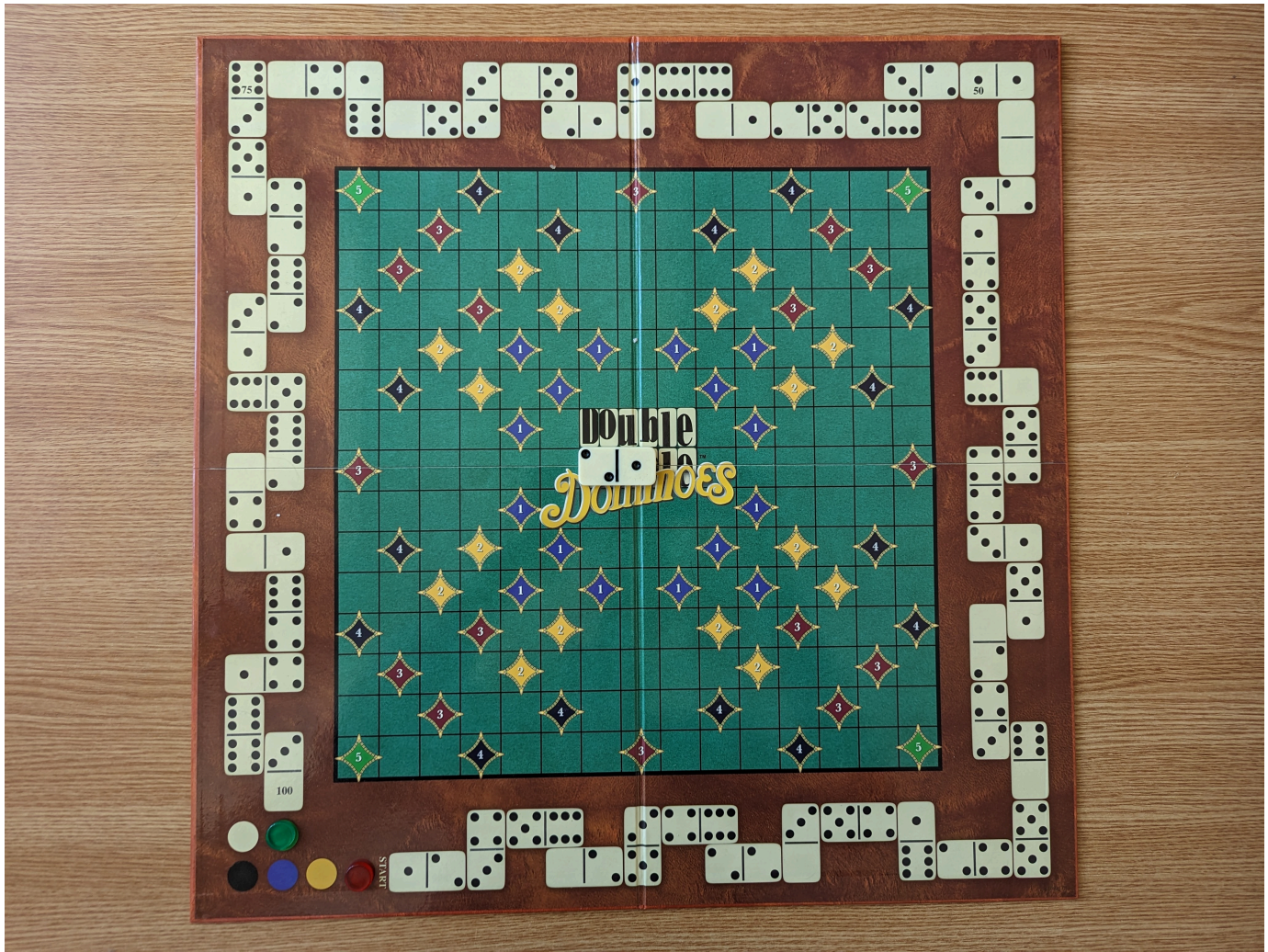
https://tinyurl.com/CV-2023-Project1

- 28 posibble domino tiles with values fromm 0-0, 0-1, …, 6-6
- 56 dominoes in total
- 2 players: each of them has 3 domino tiles



Examples of moves and scoring

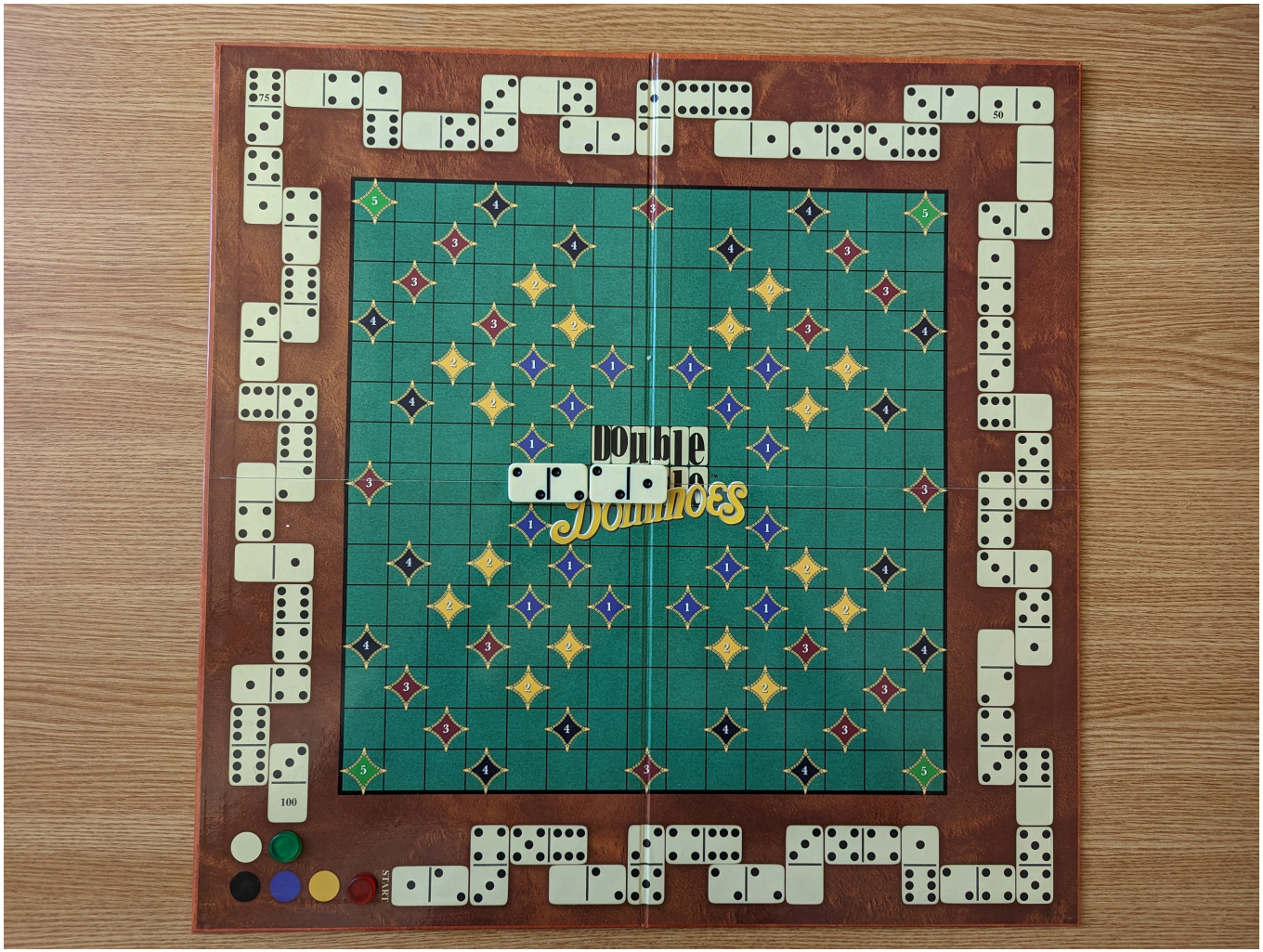
Green moves (tile 2-1 is played, nothing happens)
Score after the move: Green 0 - 0 Red



Examples of moves and scoring

Red moves (tile 2-2 is played, nothing happens)

Score after the move: Green 0 - 0 Red



Examples of moves and scoring

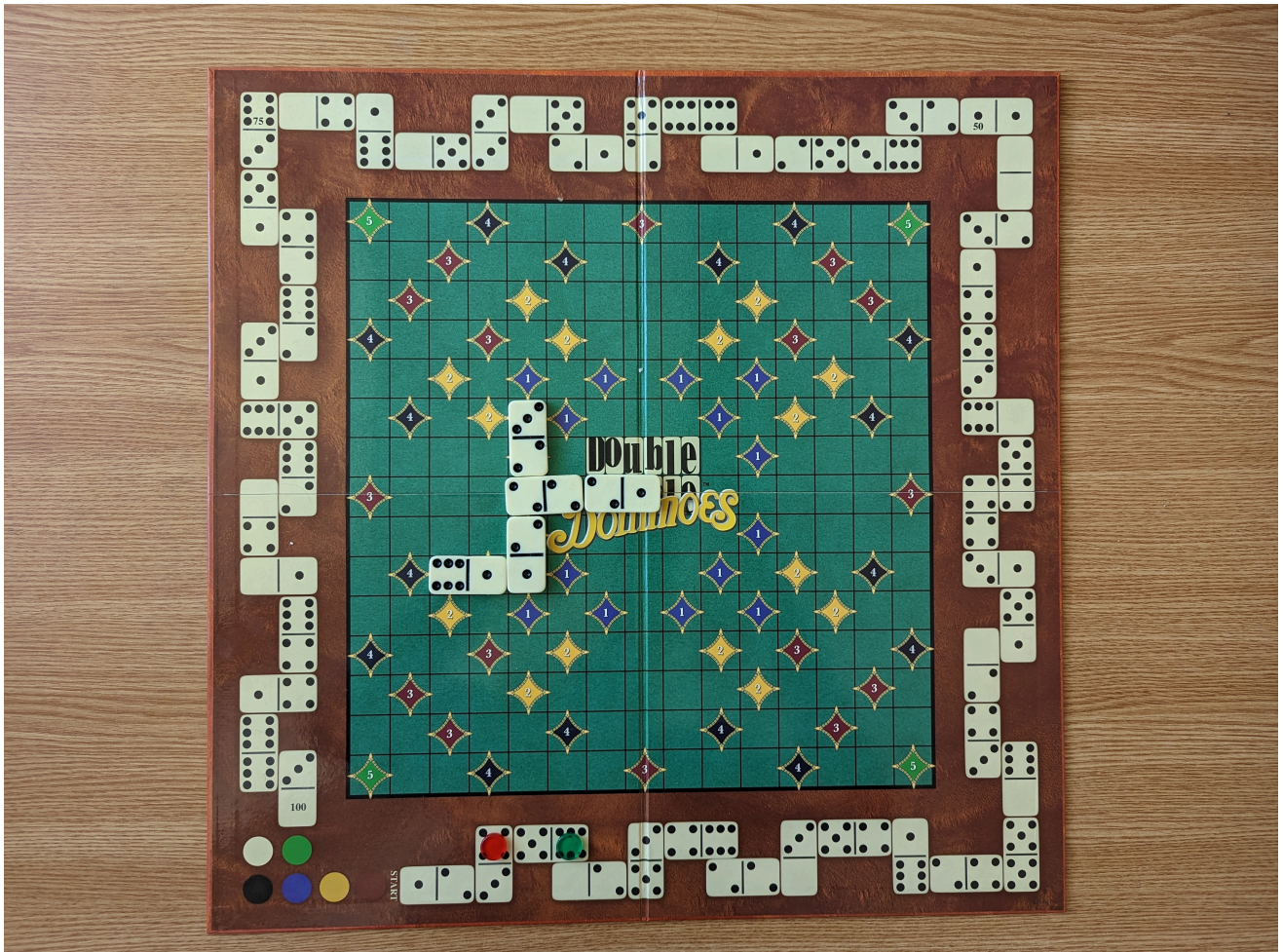
Green moves (tile 2-1 is played on the blue diamond square – 1 point)
Score after the move: Green 1 - 0 Red

Examples of moves and scoring

Red moves (tile 3-2 is played on the blue diamond square – 1 point)
Score after the move: Green 1 - 1 Red

Examples of moves and scoring

Green moves (tile 6-1 is played on the yellow diamond square – 2points + additional 3 points for each player from the score track)
Score after the move: Green 6 - 4 Red



Examples of moves and scoring

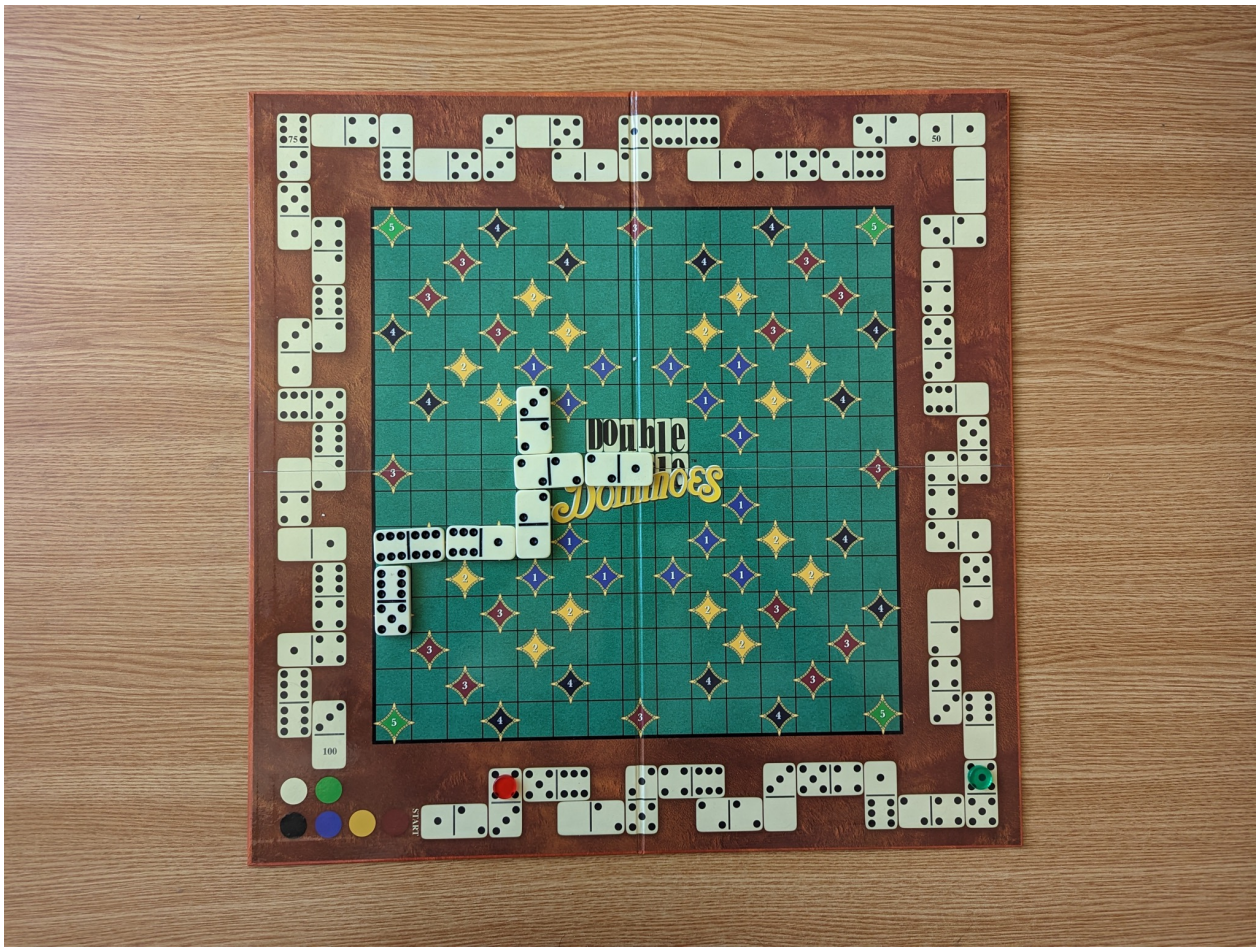
Red passes (nothing happens)

Green moves (tile 6-6 is played on the black diamond square – 2×4points + additional 3 points for green from the score track)

Score after the move: Green 17 - 4 Red

Examples of moves and scoring

Green moves (tile 6-5 is played on the black diamond square – 4points + additional 3 points for green from the score track)
Score after the move: Green 24 - 4 Red



Training data

-training data: 5 games, each game has 20 turns (1 turn = 1 image)
-training data: in total there are 100 images + 100 ground-truth annotations

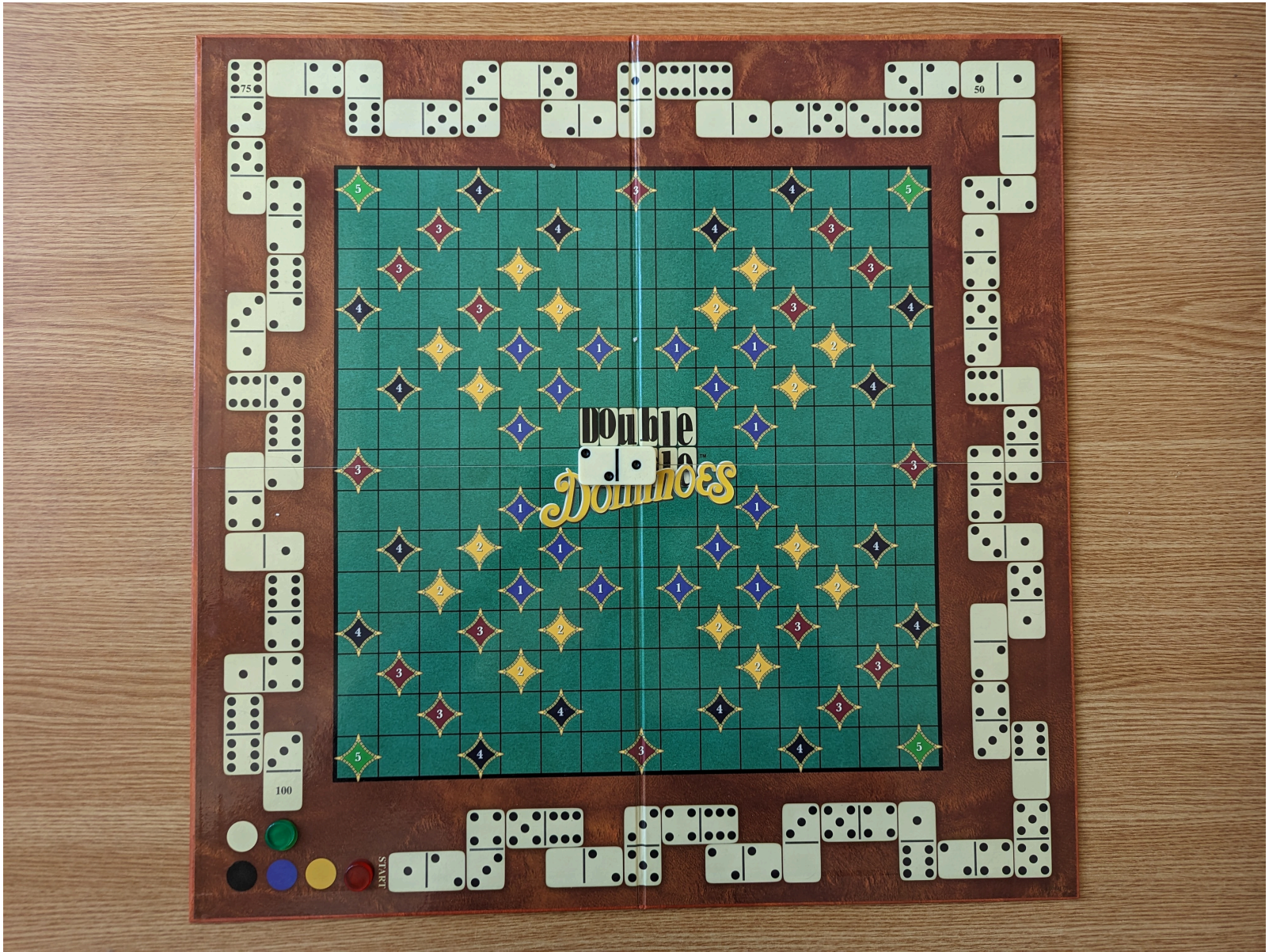
Game 1 & 2: each game contains 20 images with regular views, photos taken from above the board
1_01.jpg, 1_02.jpg, …, 1_20.jpg
2_01.jpg, 2_02.jpg, …, 2_20.jpg

Game 3: contains 20 images with rotated views, photos taken from above the board with some rotation
3_01.jpg, 3_02.jpg, …, 3_20.jpg

Game 4: contains 20 images with perspective views, photos taken by tilting the phone
4_01.jpg, 4_02.jpg, …, 4_20.jpg

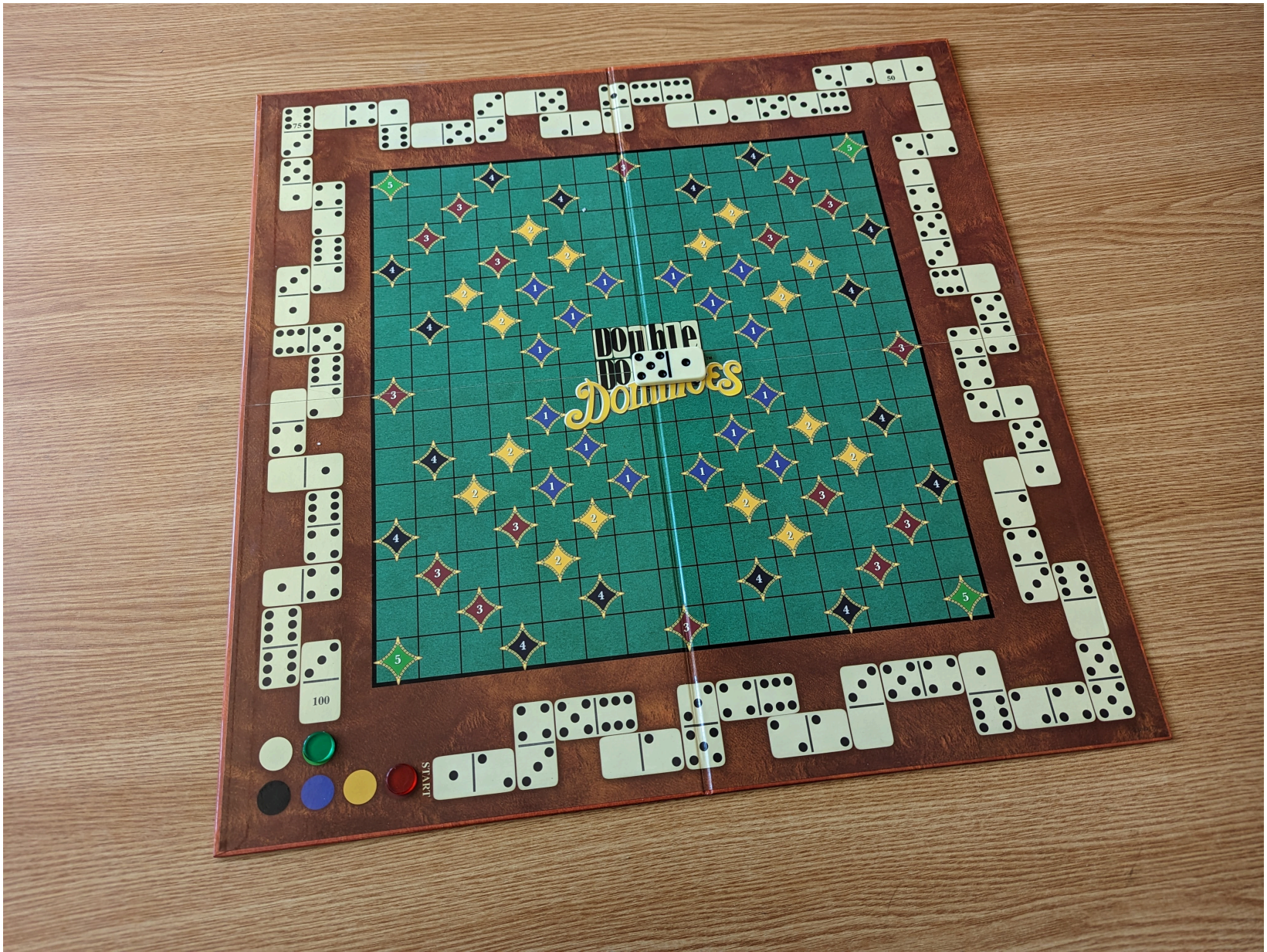
Game 5: contains 20 images with mixed view (regular, rotation, perspective)
5_01.jpg, 5_02.jpg, …, 5_20.jpg

Regular view



Rotated view

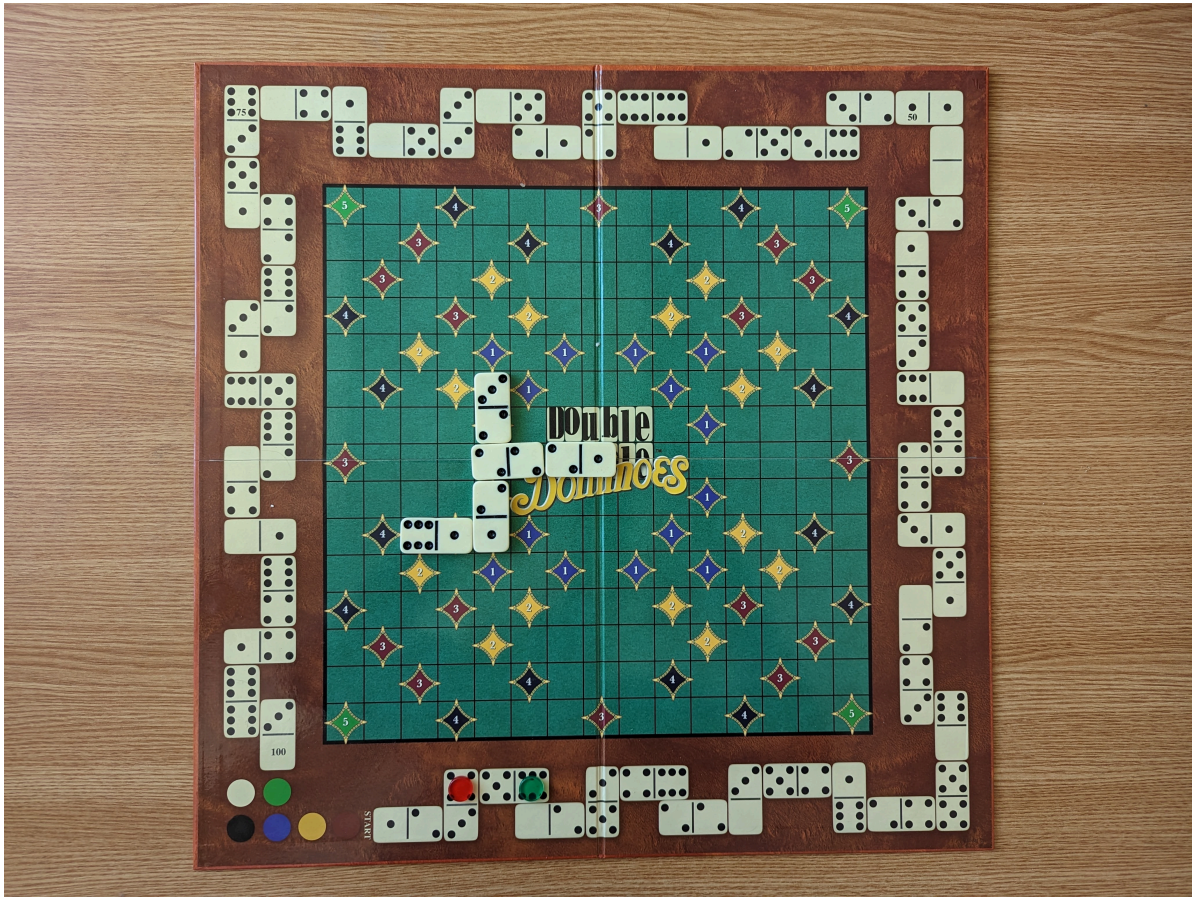
Perspective view



Ground-truth annotations

For each image we annotate:

1. the positions of the tiles added to the board (form left to right, up to down) – use notations 1-15 for rows and A-O for columns
2. the domino tile values
3. the total score at that turn by the current player



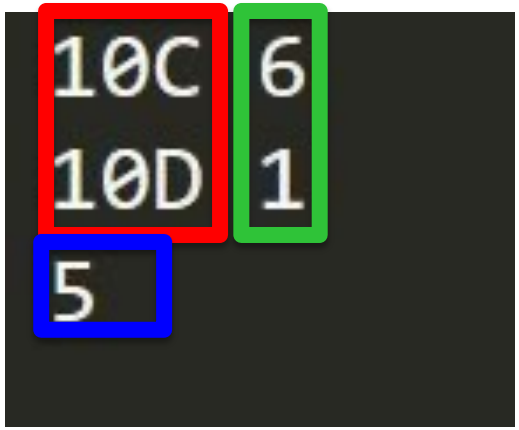

```
10C 6
10D 1
5
```

Test Data

-(almost) same as training data: 5 games, each game has 20 turns (1 turn = 1 image)
-testing data: 100 images (will be made available after the deadline for submitting the source code, on Wednesday 3rd of May)

For each test image you should produce a similar txt file (line in the annotations) containing:

1. the positions of the domino tile added to the board (form left to right, up to down) – use notations 1-15 for rows and A-O for columns
2. the domino tile values
3. the total score at that turn by the current player

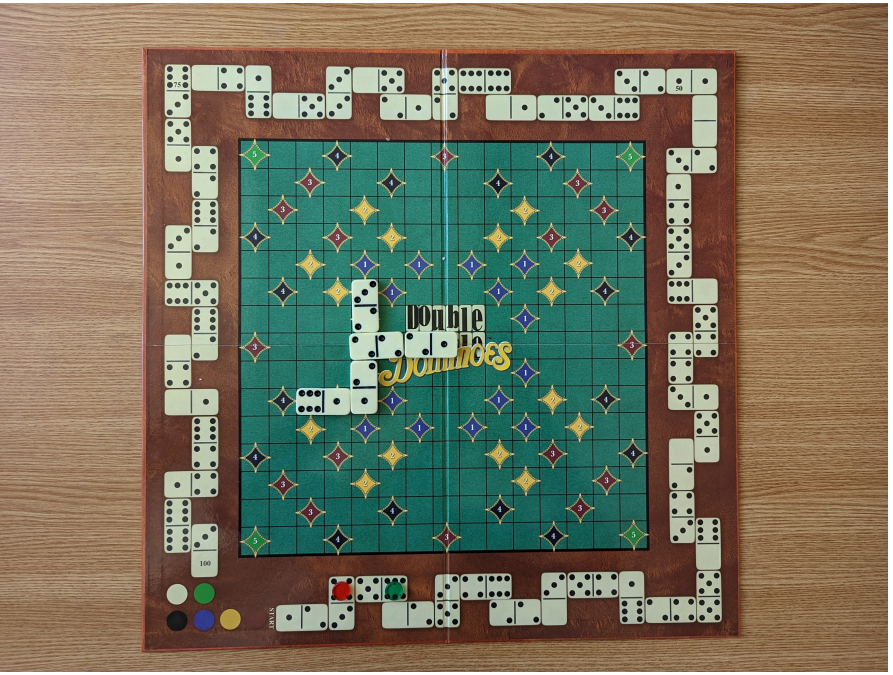


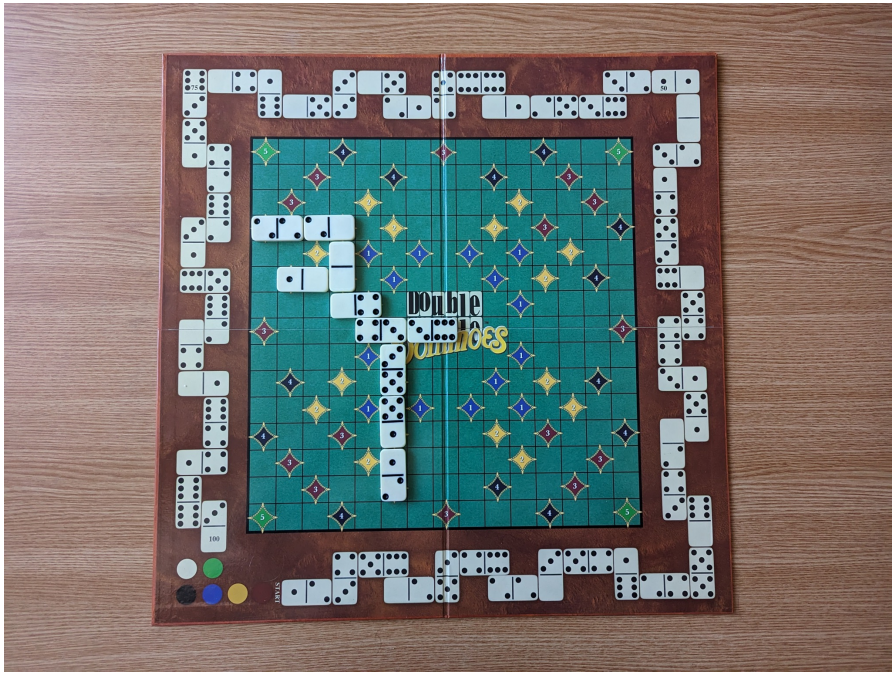
Each correctly solved image worth 0.045 points:

- correct ALL positions: 0.015 points
- correct ALL letters: 0.015 points
- correct total score: 0.015 points

Total: 100 x 0.045 = 4.5 points
Ex officio: 0.5 points (correct format)

Train vs Test data





Training images contain score pieces on the score track, testing images do no contain score pieces on the score track

Bonus task – invalid moves

```
2
8H
8I
9H
9I
9N
10N
```

Figure 8: *There are two invalid moves here, first one is placing a domino that connects with both ends forming a square, and the second one is connecting a domino that has a different number on it. We annotate all the positions that create conflict*

10 images x 0.05 points per image = 0.5 points bonus

5. Project 2
Video analysis of a snooker footage
tinyurl.com/CV-2020-project2

- **Task 1** - you receive an image of the snooker table seen from above. The task is to count the number of balls on the table and specify their color;
- **Task 2** - you receive a video containing a player making a shot. In this video the snooker table is seen from above. The task is to decide whether a ball was poted into a pocket and if so recognize the color of the poted ball and the pocket (one of the six pockets) where the ball was poted;
- **Task 3** - you receive a video containing a player making a shot. In this video the snooker table is seen from above. The task is to track the cue ball and another specified ball.
- (bonus) **Task 4** - you receive a much longer video than the previous ones (from Task 2 and Task 3). In this scenario, it may happen that several frames contain the snooker table seen from other viewpoints than above. The task here is output the score for the current footage based on the balls poted by each player. **1 point**

Task 1 – training data

Image 1

annotation

1.txt

```
10
1 white
1 black
1 pink
1 blue
1 green
1 brown
1 yellow
3 red
```

Task 1 – training data

Image 18

annotation

18.txt

```
21
1 white
1 black
1 pink
1 blue
1 green
1 brown
1 yellow
14 red
```

Task 1 – training data

Image 49

annotation

49.txt

```
21
1 white
1 black
1 pink
1 blue
1 green
1 brown
1 yellow
14 red
```

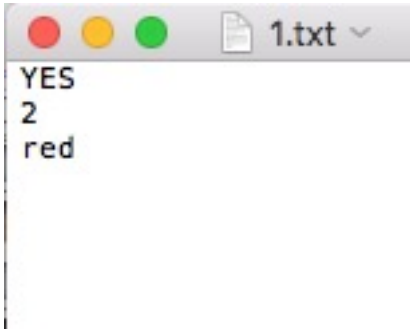
Project 2 – Task 2

- **Task 2** - you receive a test set of 25 videos containing a player making a shot. In each of the 25 videos the snooker table is seen from above. The task is to decide whether a ball was potted into a pocket and if so recognize the color of the potted ball and the pocket (one of the six pockets) where the ball was potted. By solving correctly this task for a video you will earn 0.04 points as follows: if the correct answer is NO (no ball was poted) you will receive 0.04 points if your output is NO, else you will receive 0.02 points if your output is YES, 0.01 points for specifying the correct pocket (1 - top left corner, 2 - top right corner, 3 - bottom left corner, 4 - bottom right corner, 5 - middle left corner, 6 - middle right corner) and 0.01 points for specifying the correct color of the potted ball. In each of the 25 videos there will be a maximum of one ball being potted. For a test video your output should be similar with the one provided in the directory `training_data/Task2/`. **(1 point)**

Task 2 – training data

Video 1

annotation



Task 2 – training data

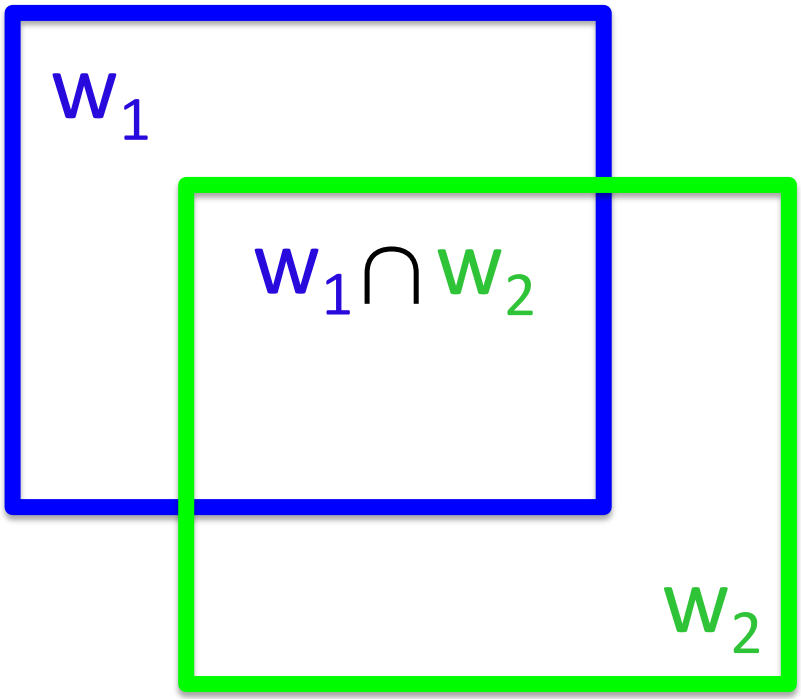
Video 2

annotation

Project 2 – Task 3

- **Task 3** - you receive a test set of 25 videos containing a player making a shot. In each of the 25 videos the snooker table is seen from above. The task is to track the cue ball (the white ball) and another specified ball. The initial bounding boxes of the two balls to be tracked are provided for the first frame. By correctly solving the tracking problem for a video you will earn 0.04 points (0.02 points for each correctly tracked ball). In each video we will consider that your algorithm *correctly tracks a ball* if in more (greater or equal) than 80% of the frames your algorithm *correctly localizes the ball to be tracked*. We consider that your algorithm correctly localizes the ball to be tracked in a specific frame if the value of the IOU (intersection over union) beetween the window provided by your algorithm and the ground-truth window is more than 20%. For a test video your output should be similar with the one provided in the folder `training_data/Task3/`. **(1 point)**

Intersection over union



$$IOU(w_1, w_2) = \frac{area(w_1 \cap w_2)}{area(w_1 \cup w_2)}$$

$$IOU(w_1, w_2) = \frac{area(w_1 \cap w_2)}{area(w_1) + area(w_2) - area(w_1 \cap w_2)}$$

Task 3 – training data

Video 1

annotation

1_white.txt

```
196 -1 -1 -1 -1
0 859 519 890 550
1 863 522 888 550
2 859 520 889 544
3 861 522 890 549
4 860 521 886 547
5 860 521 888 547
6 858 518 893 549
7 862 520 888 546
8 860 520 887 547
9 861 521 889 548
10 860 521 887 547
```

```
178 852 98 873 120
179 853 99 874 121
180 853 99 874 121
181 854 99 875 121
182 854 99 875 121
183 855 99 876 121
184 855 99 876 121
185 855 99 876 121
186 856 99 877 121
187 856 99 877 121
188 855 99 876 121
189 856 99 877 121
190 856 99 877 121
```

Task 3 – training data

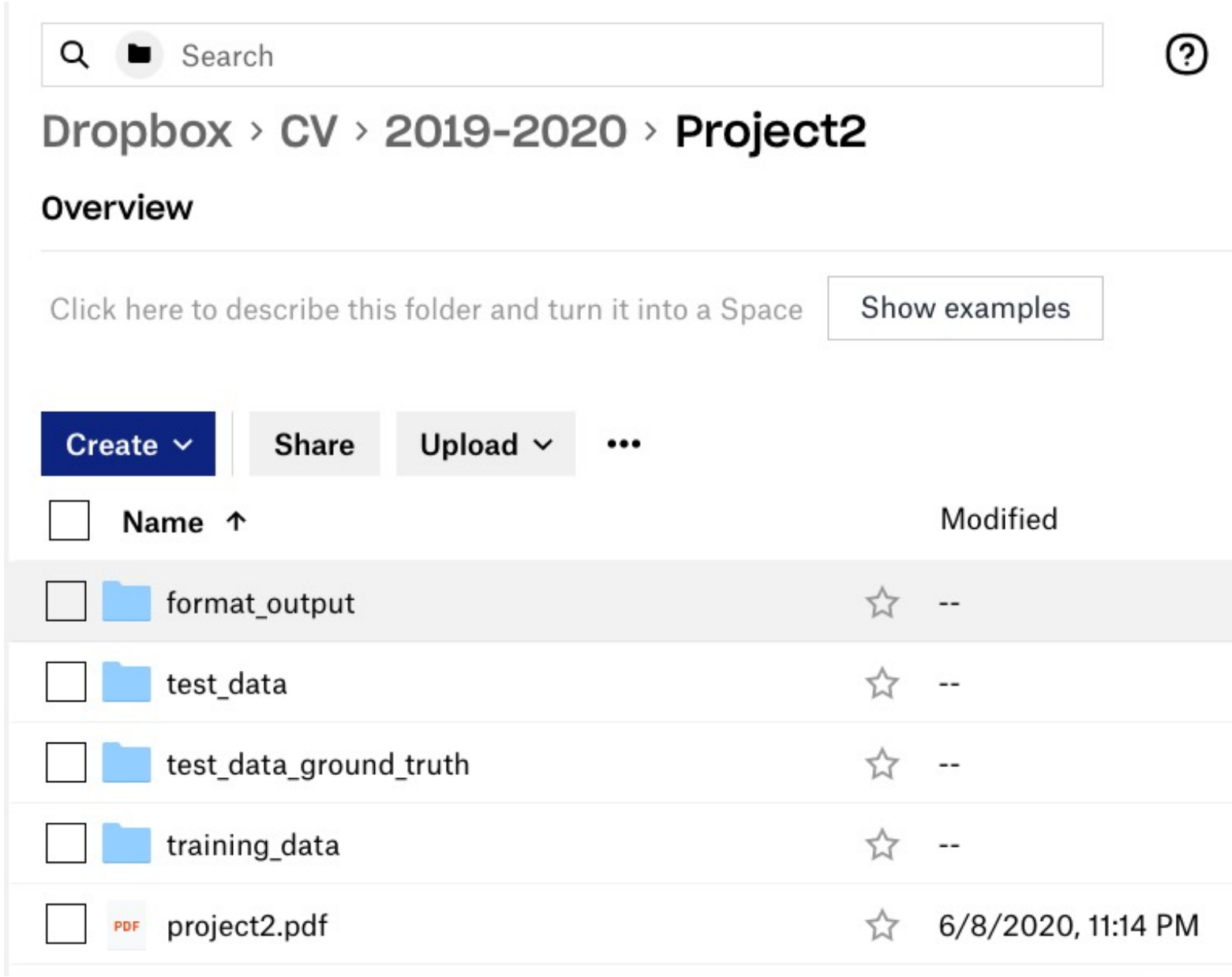
Video 2

Project 2 – Task 4

- (bonus) **Task 4** - - you receive a test set of 25 videos containing a player making a shot. Different than the previous tasks, in this videos the snooker table can be filmed from different viewpoints (not only from above). The task is to track the cue ball. The task here is much harder as you have diferent scales of the cue ball, changes in camera viewpoint, the initial bounding box of the white ball is not provided. By correctly solving the tracking problem for a video you will earn 0.04 points. The rules described at Task 3 apply here, meaning that your algorithm should correctly localizes the cue ball in more than 80% of the frames, at each frame correctly localization means that the window provided by your algorithm should have IOU more than 20% with respect to the ground-truth window. For a test video your output should be similar with the one provided in the folder `training_data/Task4/` **(1 point)**

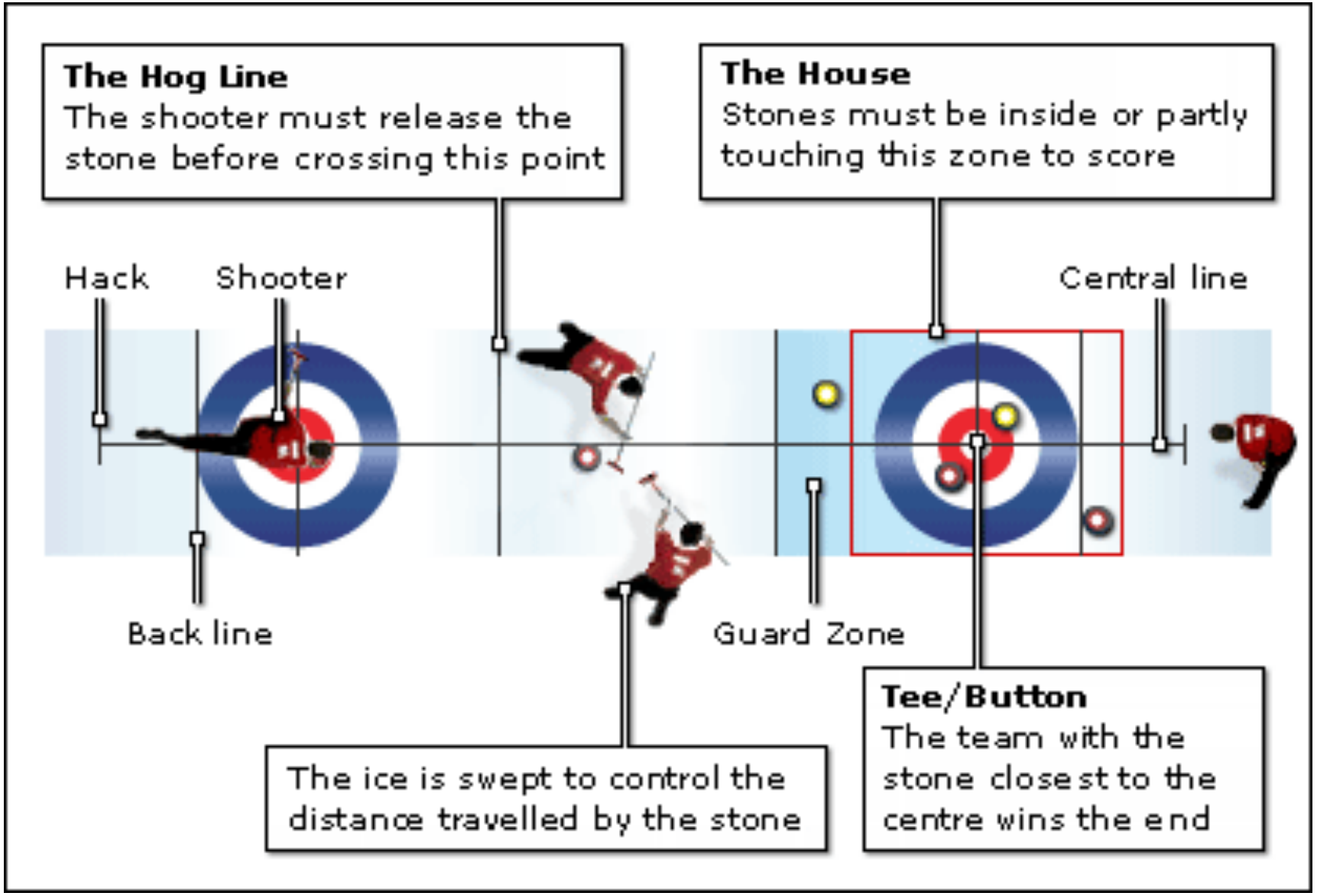
Task 4 – training data

Project 2: tinyurl.com/CV-2020-project2



6. Project 2 – Video analysis of footages from the sport of curling - tinyurl.com/CV-2021-Project2

Project 2 – Video analysis of footages from the sport of curling



Task 1 – count the stones in a frame

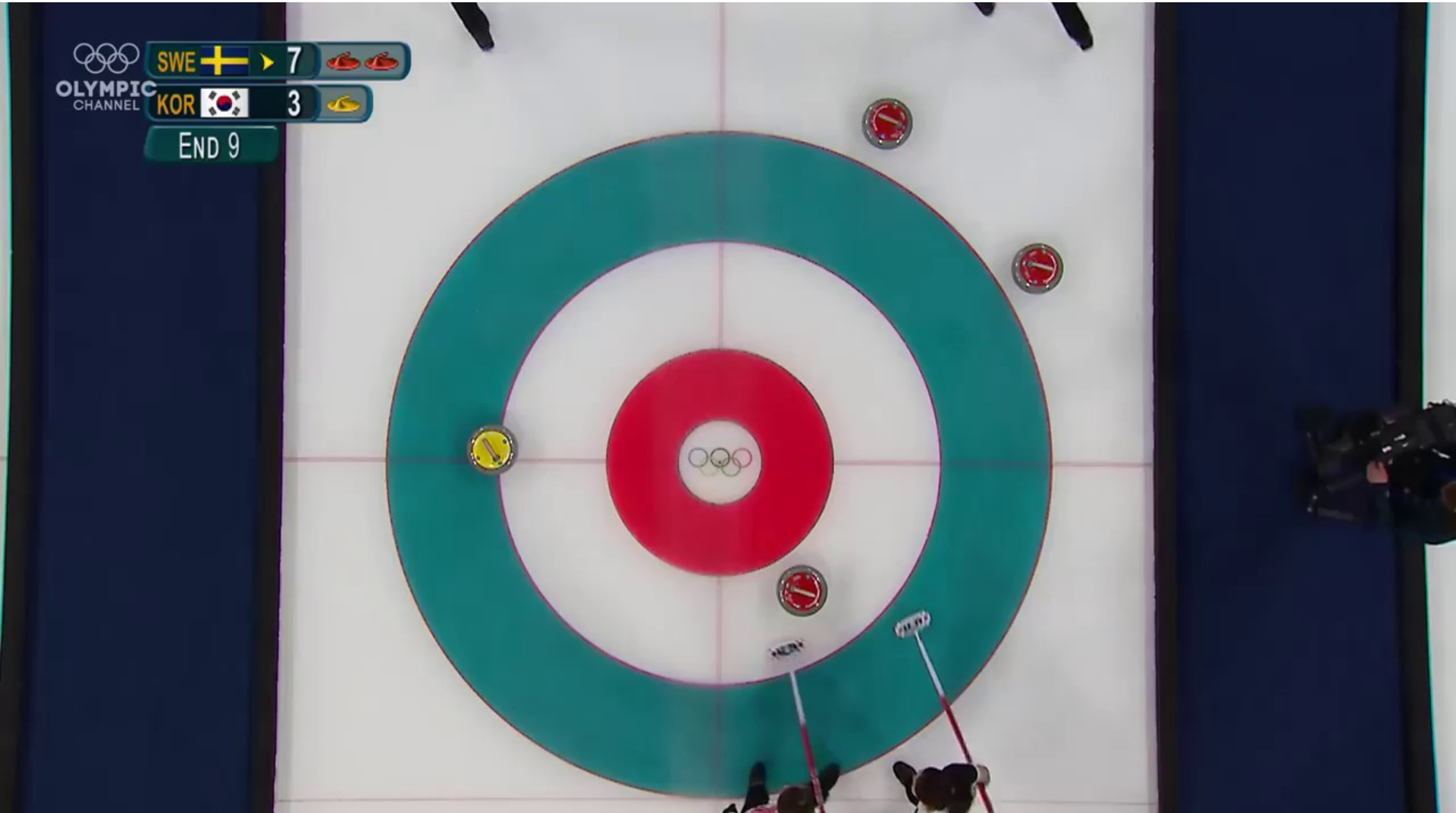
- the task is to count number of stones in an image and specify how many are red and how many are yellow

- you are given 25 images for training (with annotated ground-truth)

- at test time you have to do the prediction on 25 test images

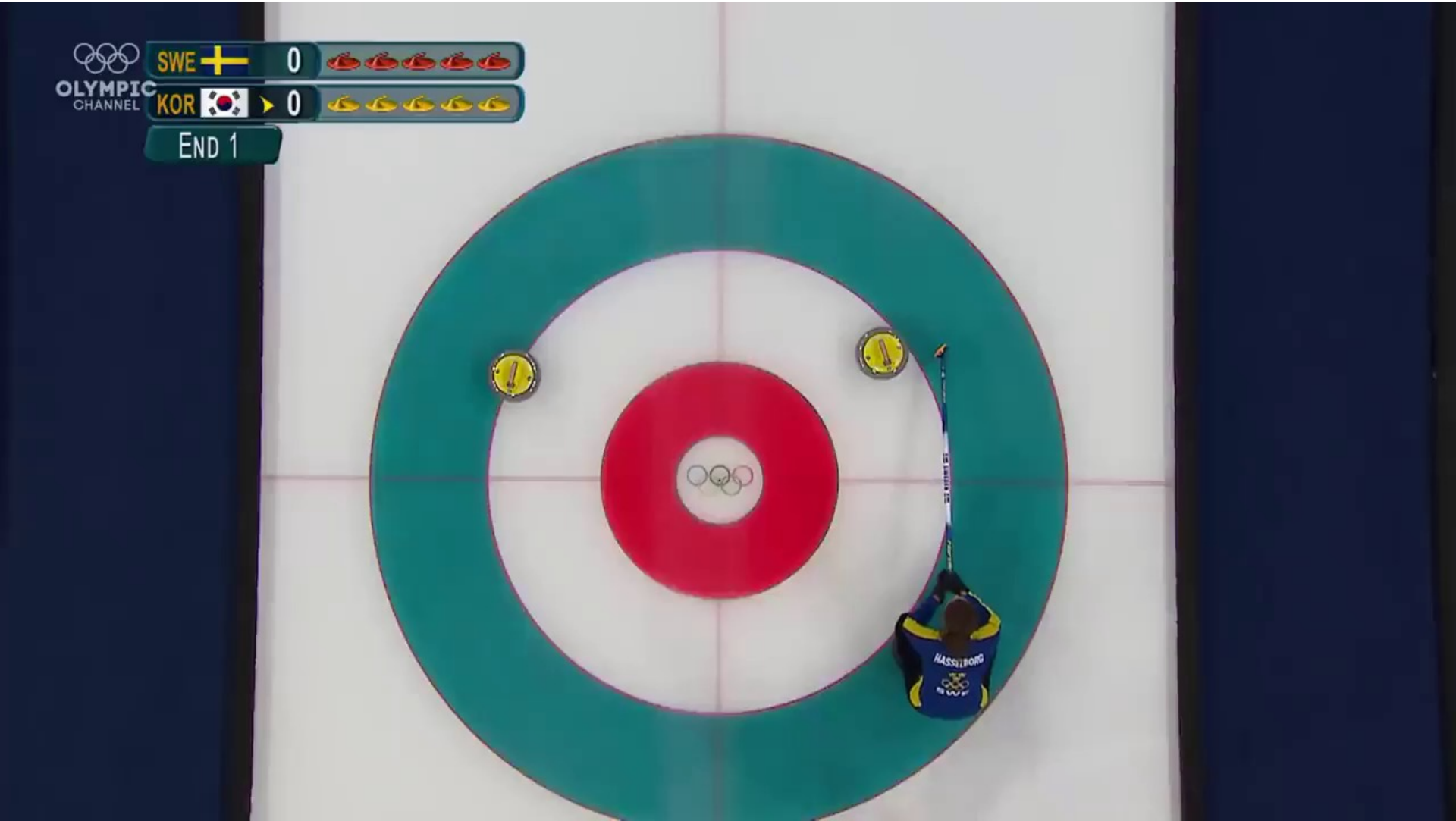
Task 1 – count the stones in a frame

Training example 1: 4 stones (3 red + 1 yellow)



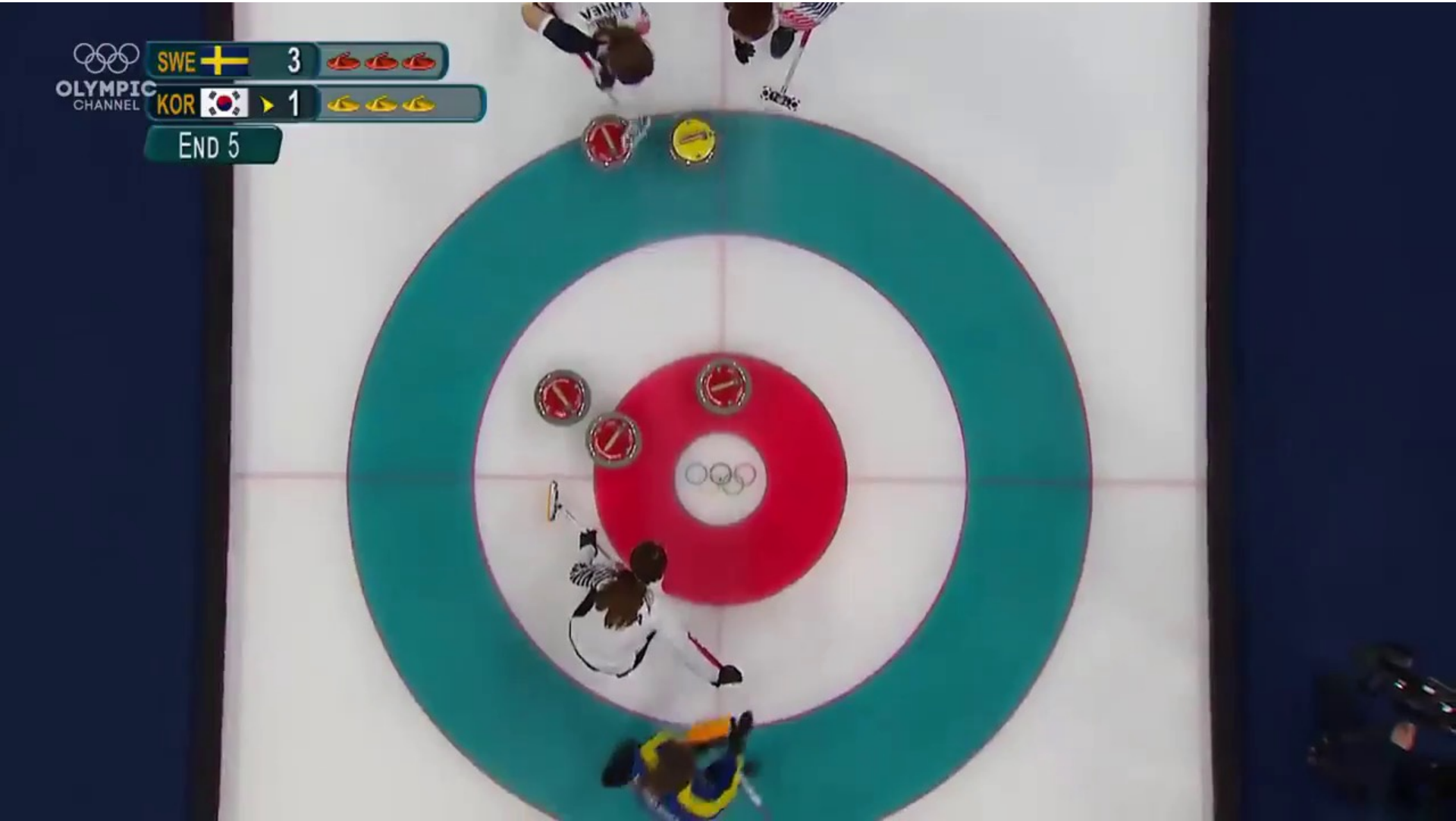
Task 1 – count the stones in a frame

Training example 2: 2 stones (0 red + 2 yellow)



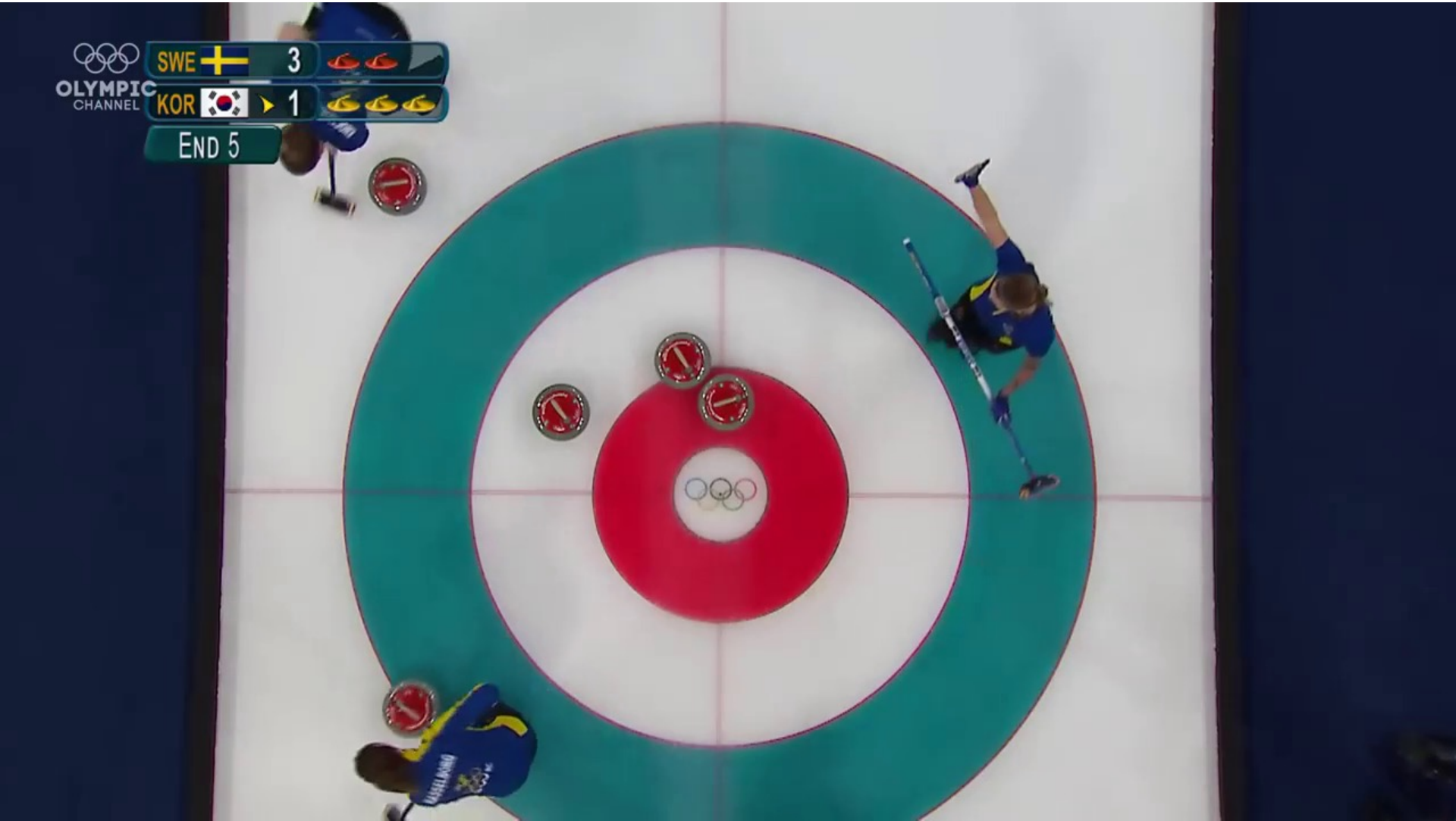
Task 1 – count the stones in a frame

Training example 4: 5 stones (4 red + 1 yellow)



Task 1 – count the stones in a frame

Training example 22: 5 stones (5 red + 0 yellow)



Task 2 – infer the score after a play

- the task is to infer the score after a play (what is the score at the end of the video)

- how to compute the score:
  - determine how many red and yellow stones are in the House (green circle). If there is no stone in the House the score is 0-0
  - only one color makes points, the one having the closest stones to the center (number of points = # of this stones)

Task 2 – infer the score after a play

Training example 10: 3 red vs 0 yellow

Task 2 – infer the score after a play

Training example 14: 1 red vs 0 yellow

Task 3 – track a released stone

- the task is to track the released stone (you are given the initial bounding box in the first frame)

- the scene is constrained, the ice surface is seen always from above

- compute at each frame the detection, compare it for evaluation with the ground truth

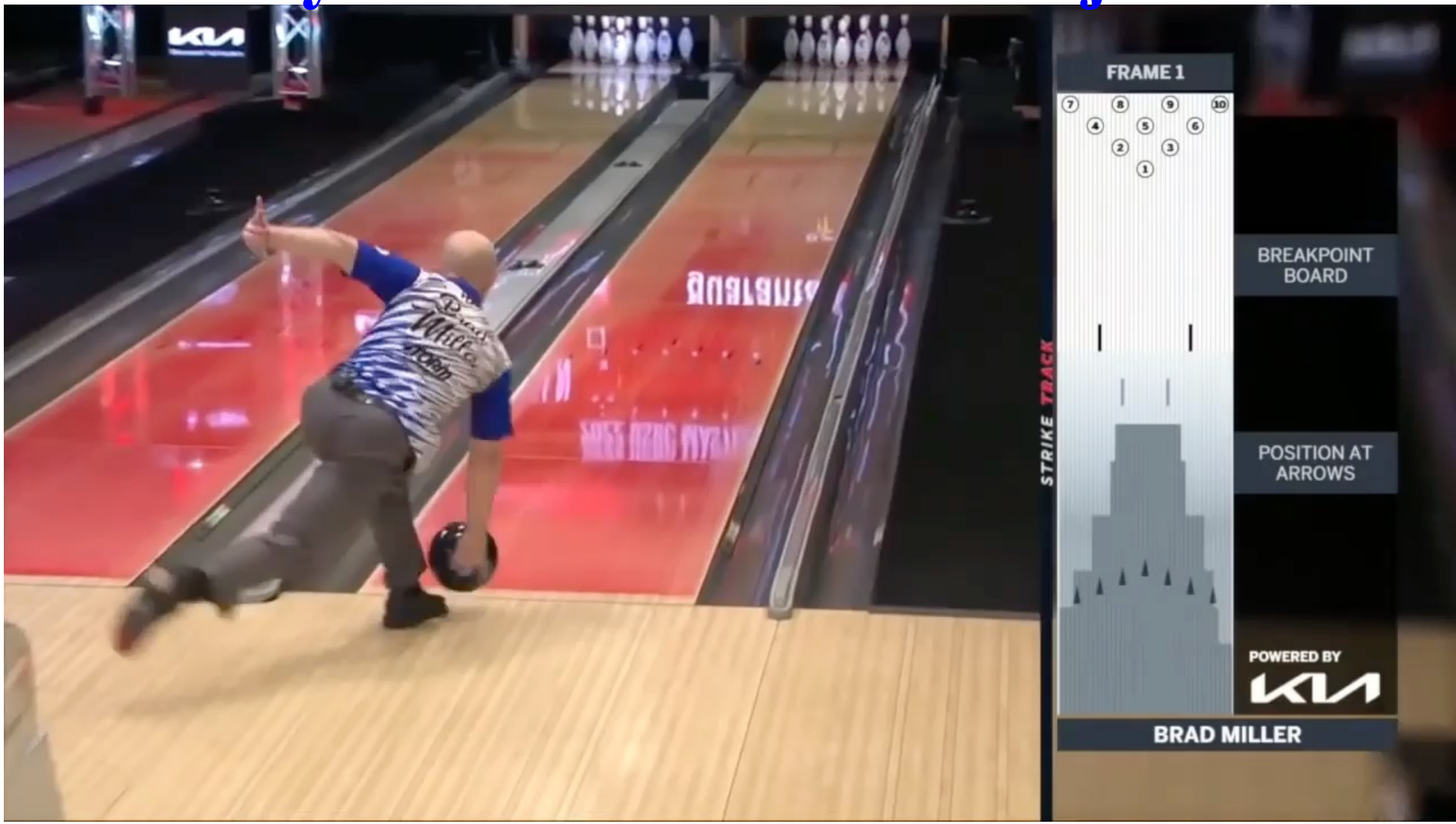
Task 3 – track a released stone

Training example 3

Task 4 – track a released stone (unconstrained)

7. Video analysis of bowling

tinyurl.com/CV-2022-Project2

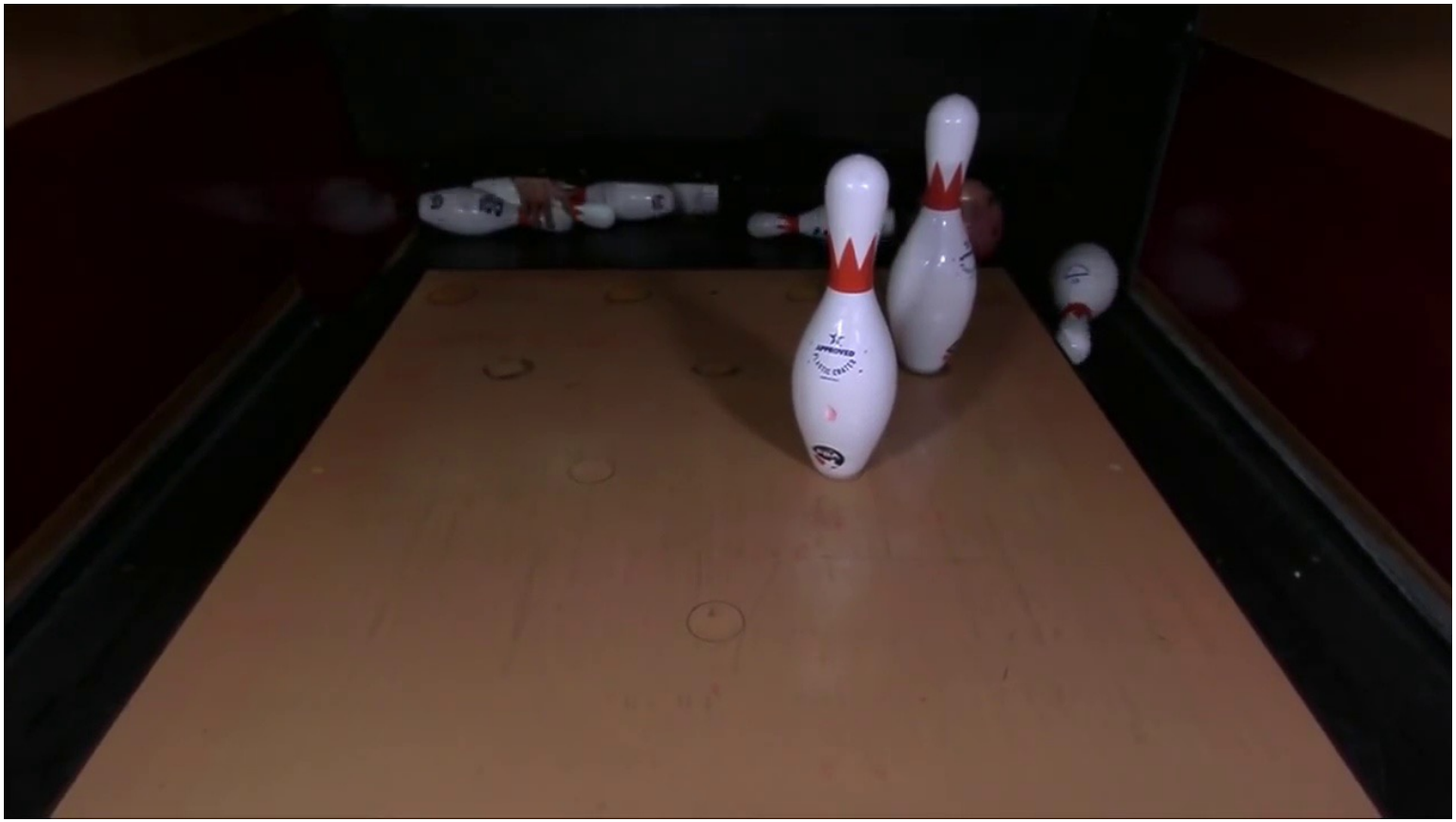


Task 1 – full configurations templates

We provide images (templates) with full configurations for both lanes

Task 1 – count the (standing) pins + configuration

Training example 1: 2 pins + configuration 3-6

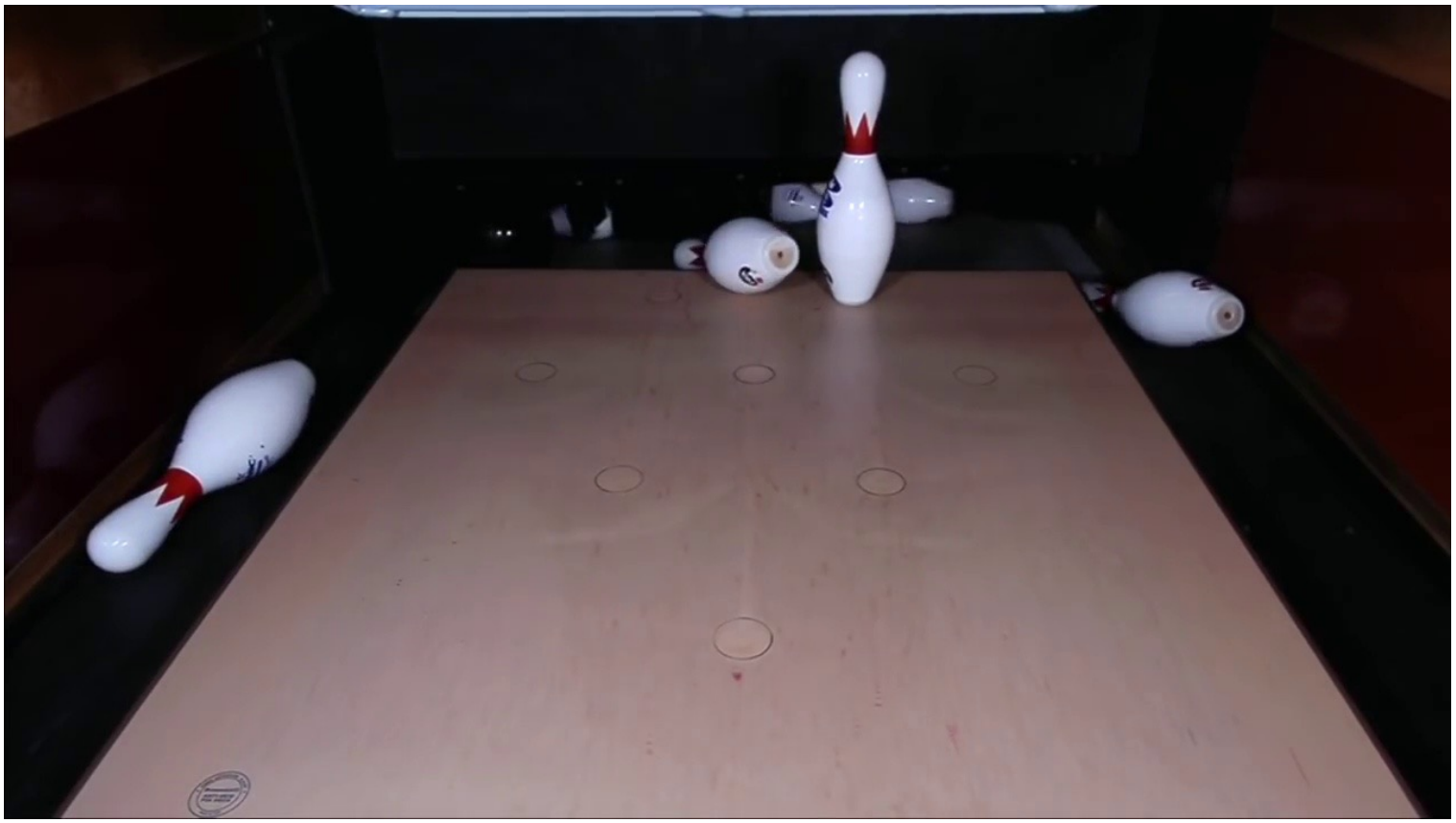


1.txt

1	2
2	3
3	6

Task 1 – count the pins + configuration

Training example 2: 1 pin + configuration 9

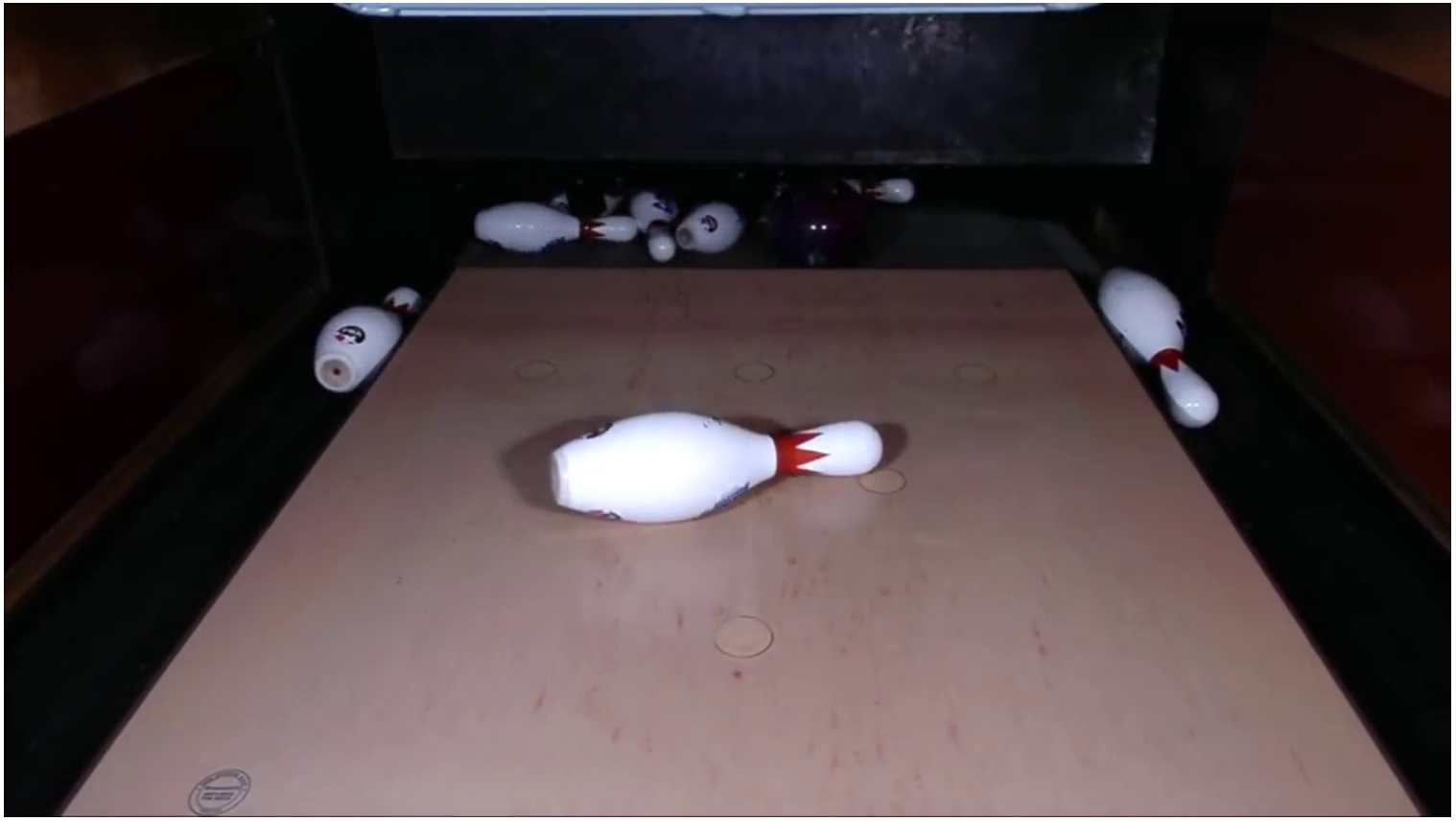


2.txt

1	1
2	9

Task 1 – count the pins + configuration

Training example 3: 0 pins + no configuration



3.txt

```
1  0
2
3
```

Task 1 – count the pins + configuration

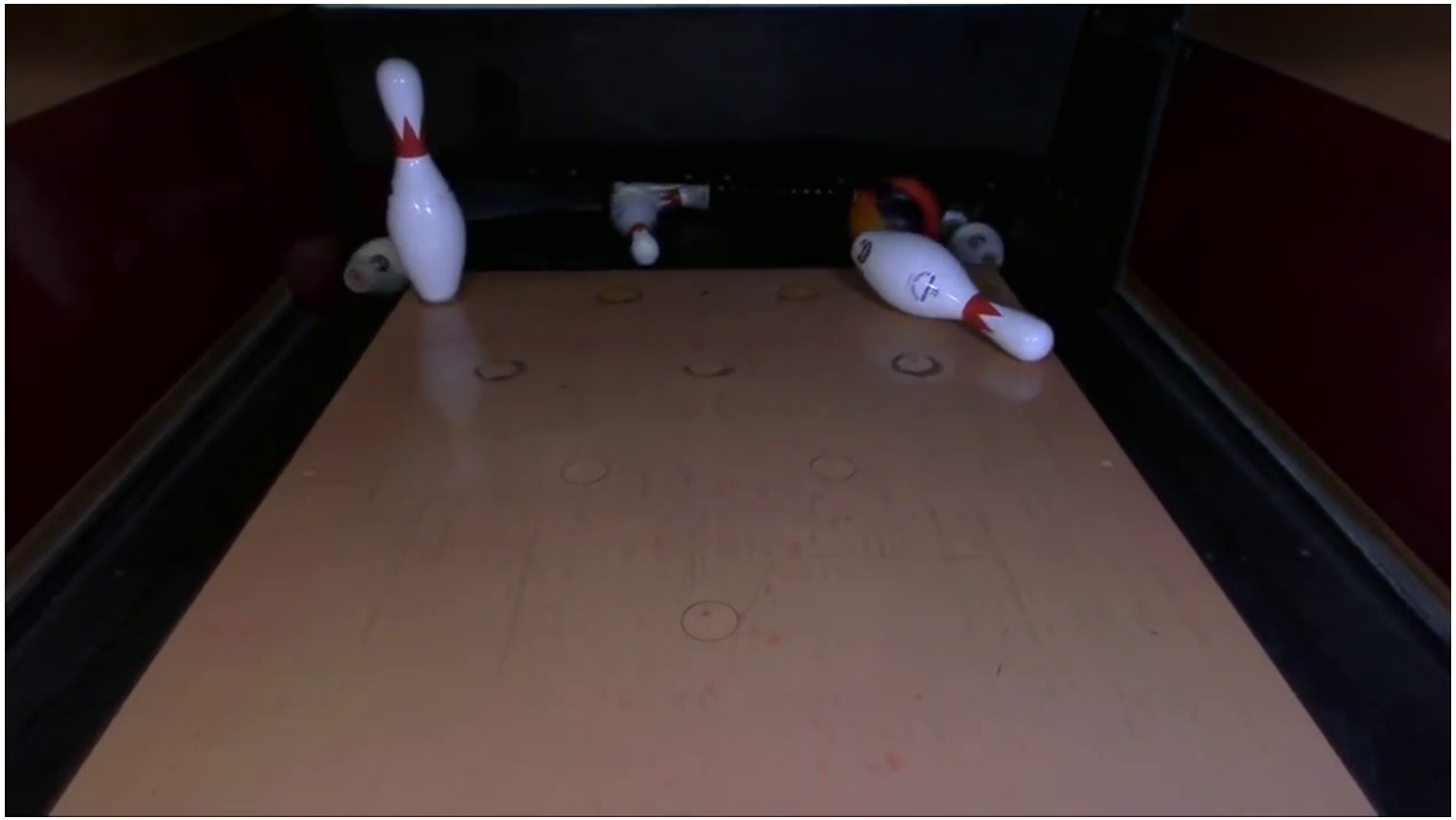
Training example 6: 4 pins + configuration 1-2-8-10

6.txt

```
1  4
2  1
3  2
4  8
5  10
```

Task 1 – count the pins + configuration

Training example 22: 1 pin + configuration 7



22.txt

1	1
2	7

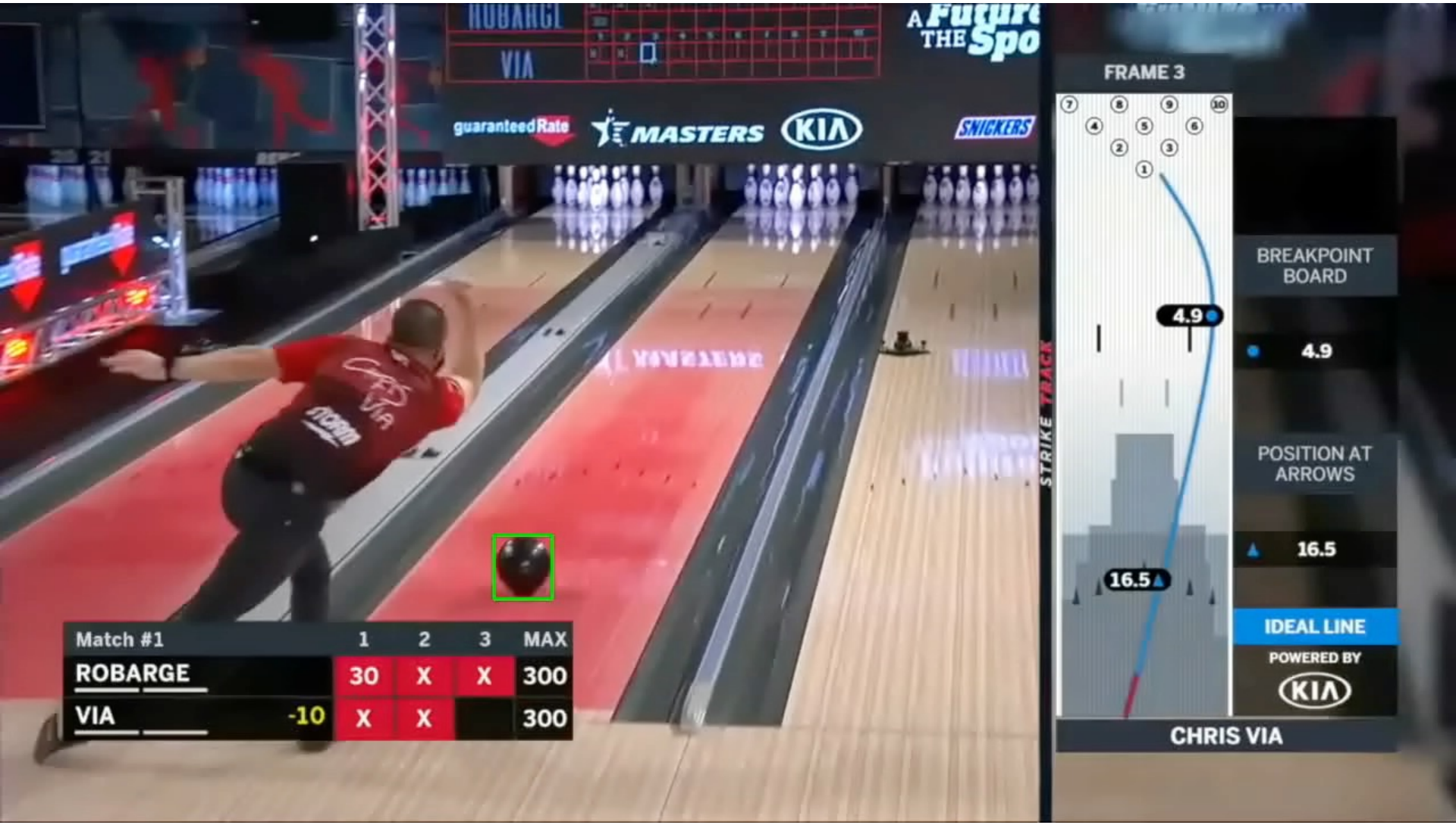
Task 2 – track the bowling ball

- the task is to track the bowling ball in a given video (you are given the initial bounding box in the first frame of the video)

- compute at each frame the detection, compare it for evaluation with the ground truth

Task 2 – track a released stone

Training example 2

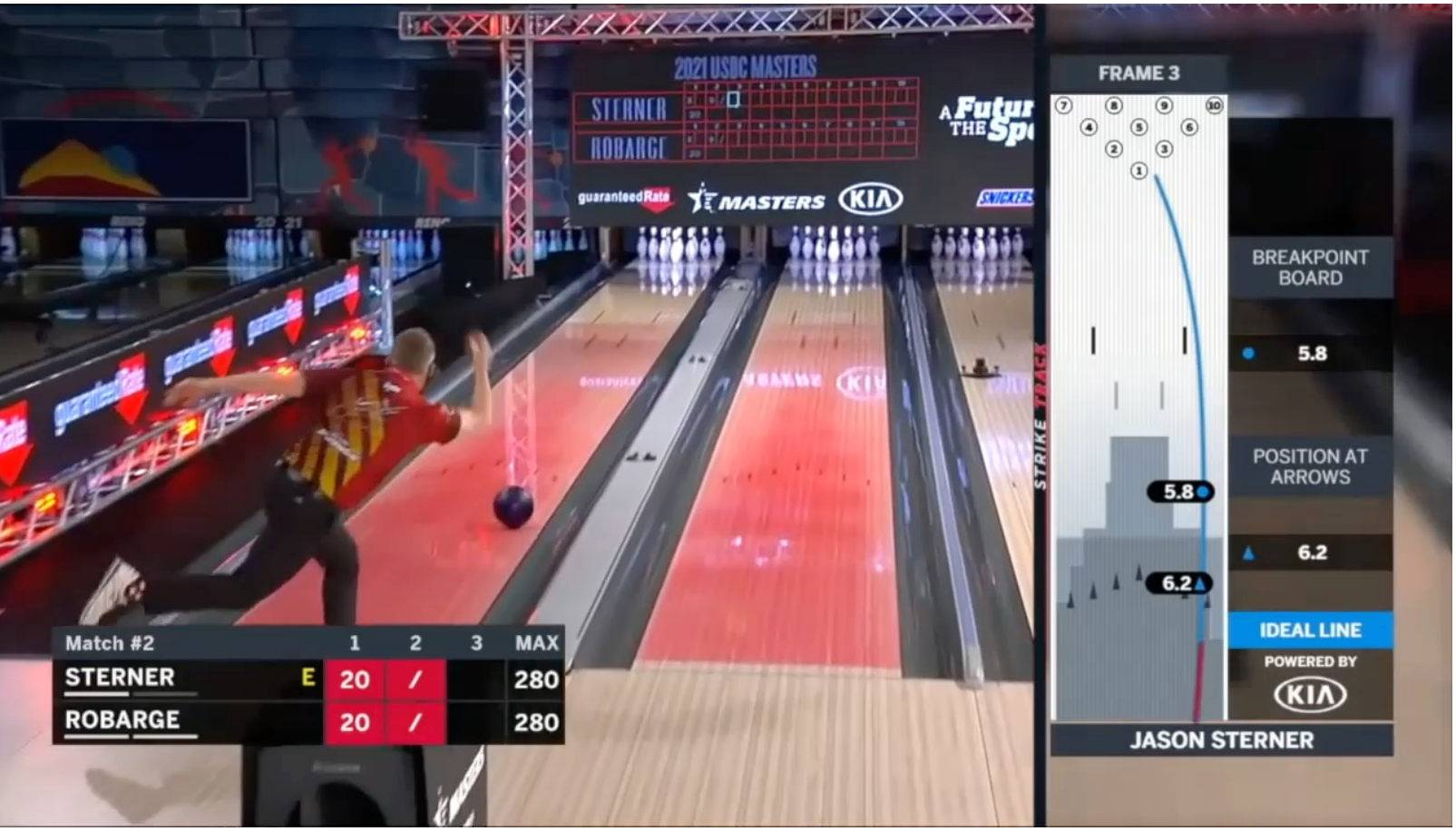


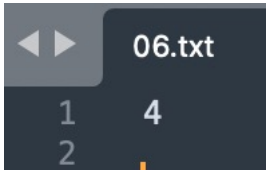
Task 3 – #standing pins after bowling

- the task is to count the number of standing pins after bowling (after the bowling ball hits the pins and some of them are knocked down but some of them may be standing)

Task 3 – #standing pins after bowling

Training example 6

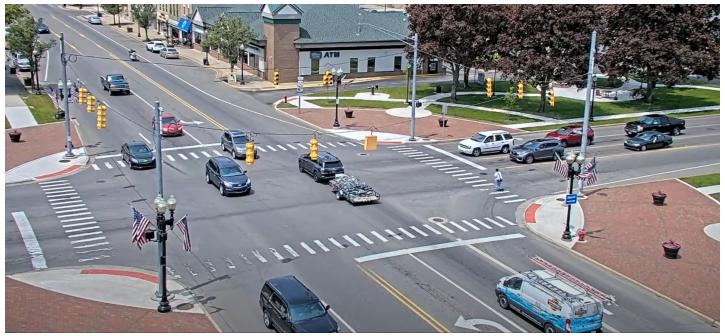


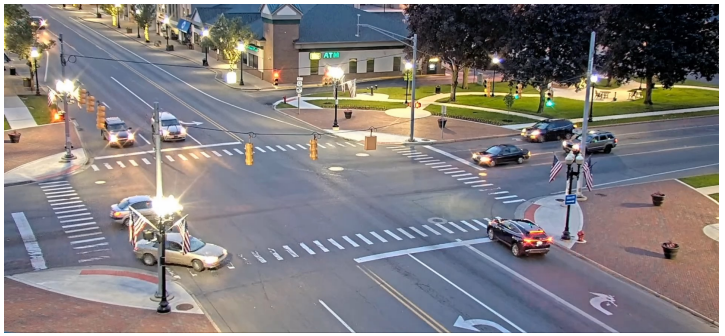


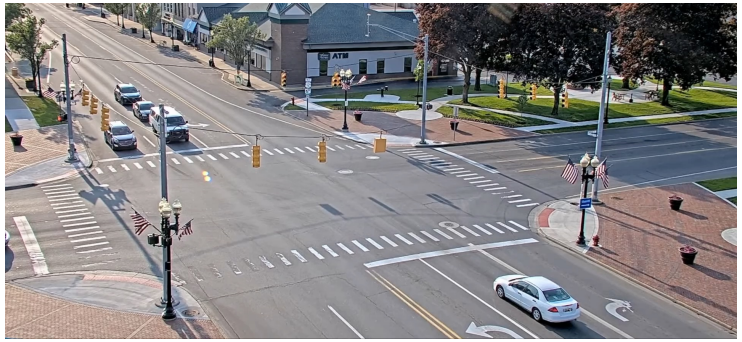
8. Visual traffic monitoring at a road intersection

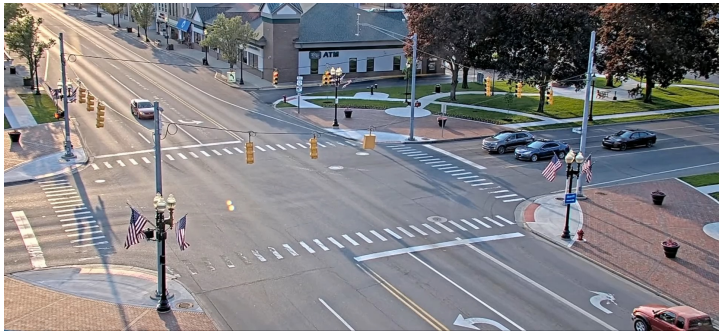
tinyurl.com/CV-2023-Project2

- Static camera monitoring the traffic in an intersection
- Videos collected at different times during day/week
- Changes in illumination



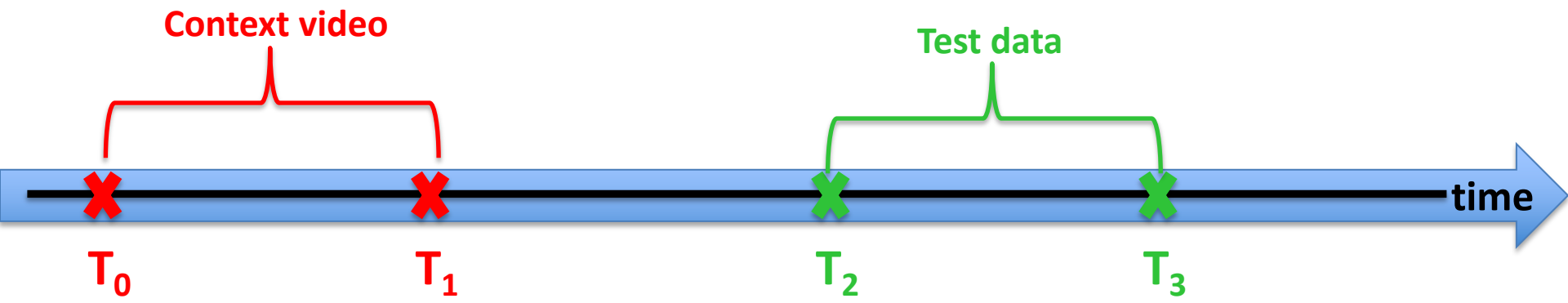






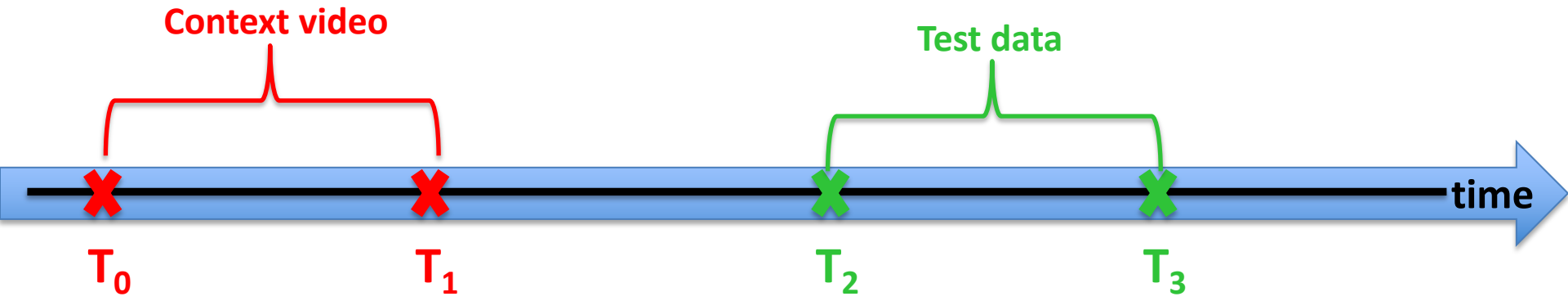
Tasks

For each task you are given a “context” video that is taken with a few minutes before the test data (images or video) that you have to analyze.



Tasks

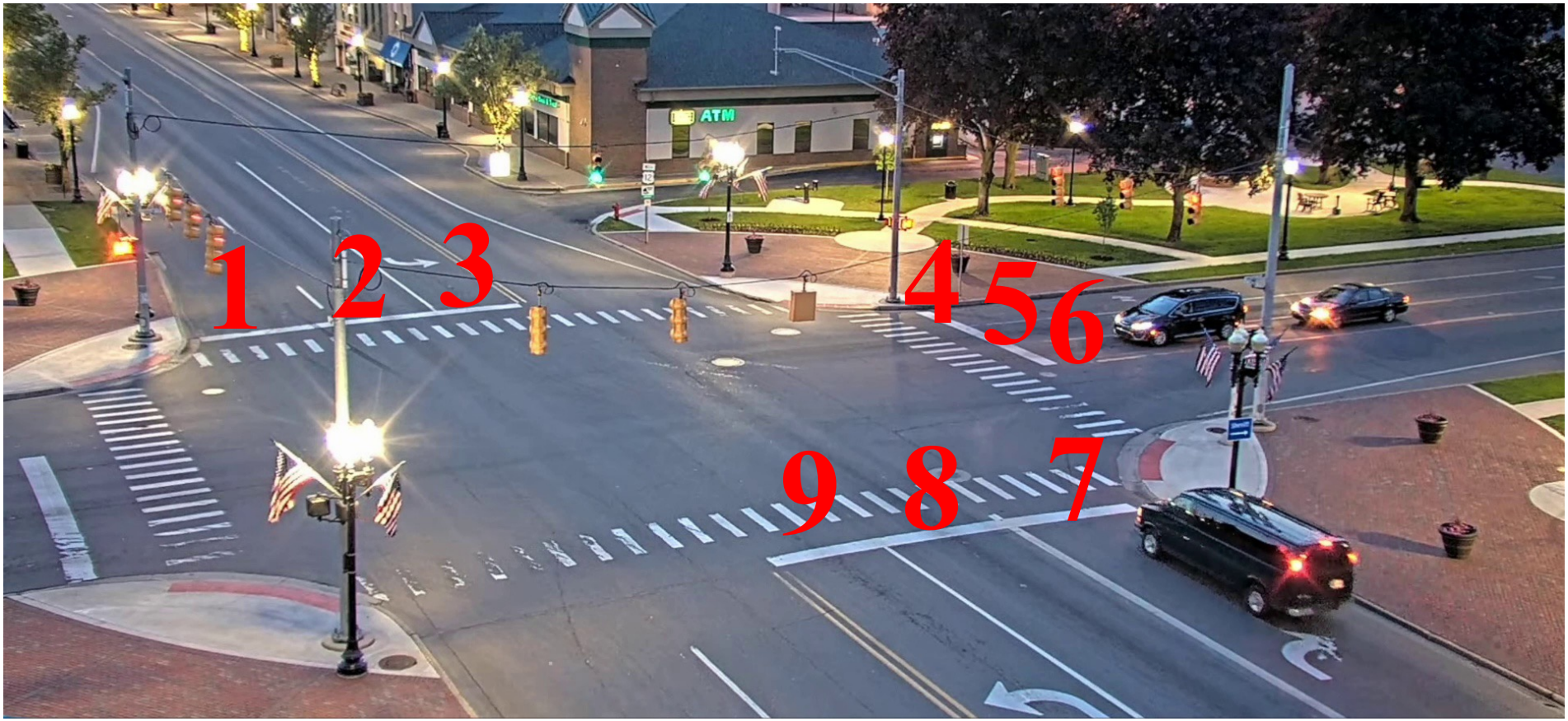
For each task you are given a "context" video that is taken with a few minutes before the test data (images or video) that you have to analyze.



1. Classifies road lanes as being occupied or not
2. Track a specific vehicle in a video
3. Count the trajectories in a video

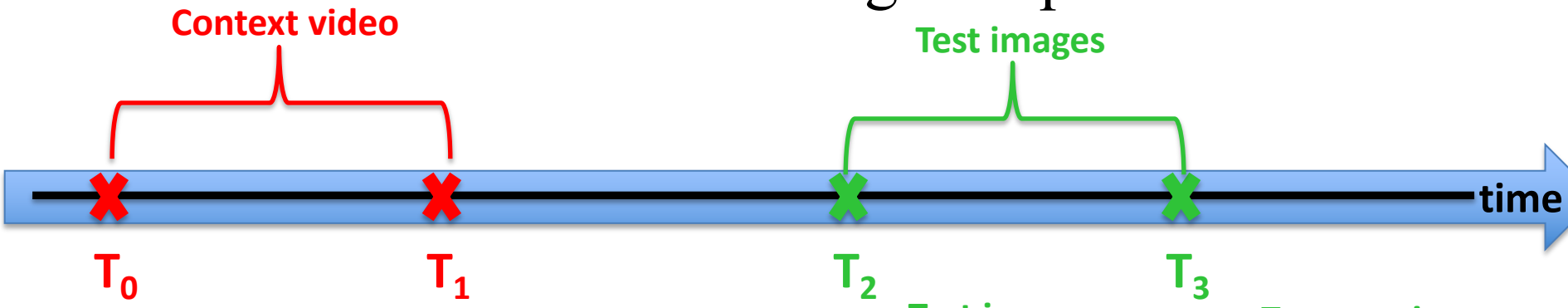
Task1 - Lane numbering

1. Classifies road lanes as being occupied or not

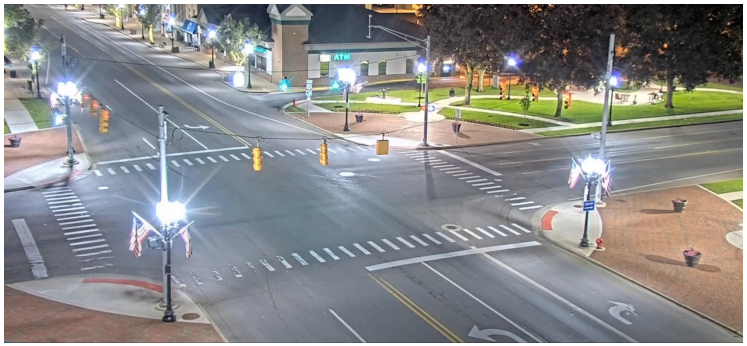


Task1

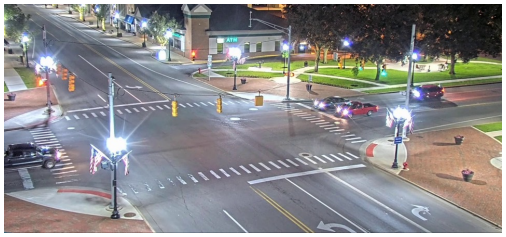
1. Classifies road lanes as being occupied or not

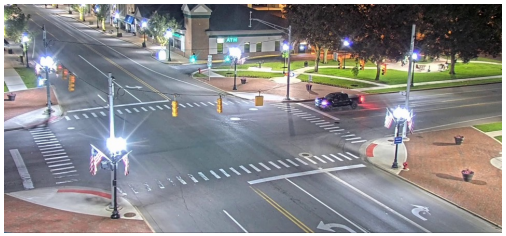


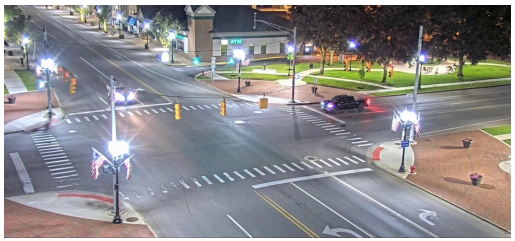
Context video



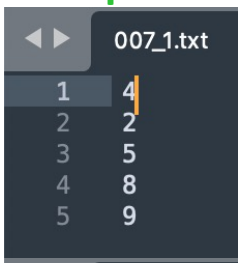
Test images

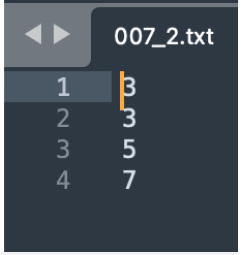


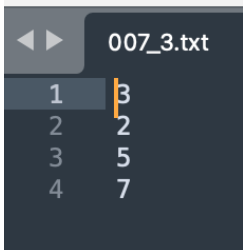




Test queries



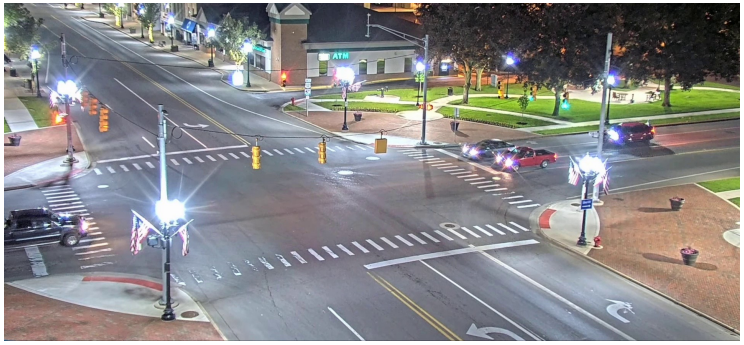


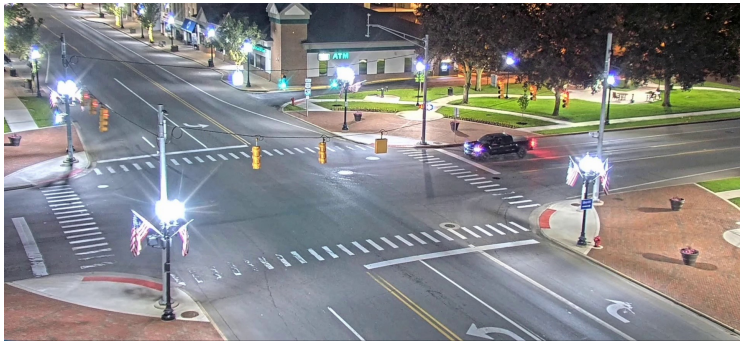


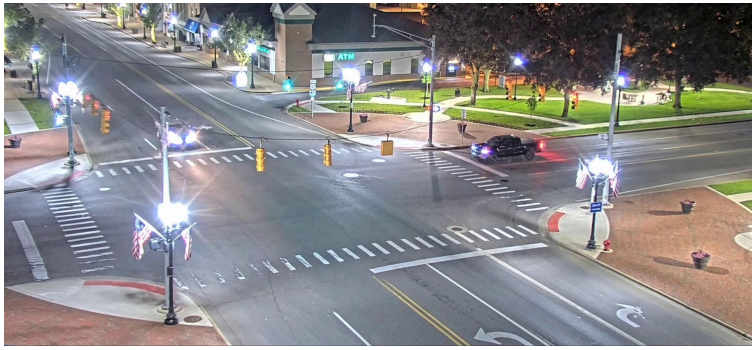
Task1

1. Classifies road lanes as being occupied or not

Test images







Test queries

007_1.txt

```
4
2
5
8
9
```

007_2.txt

```
3
3
5
7
```

007_3.txt

```
3
2
5
7
```

007_1.txt

```
4
2 0
5 1
8 0
9 0
```

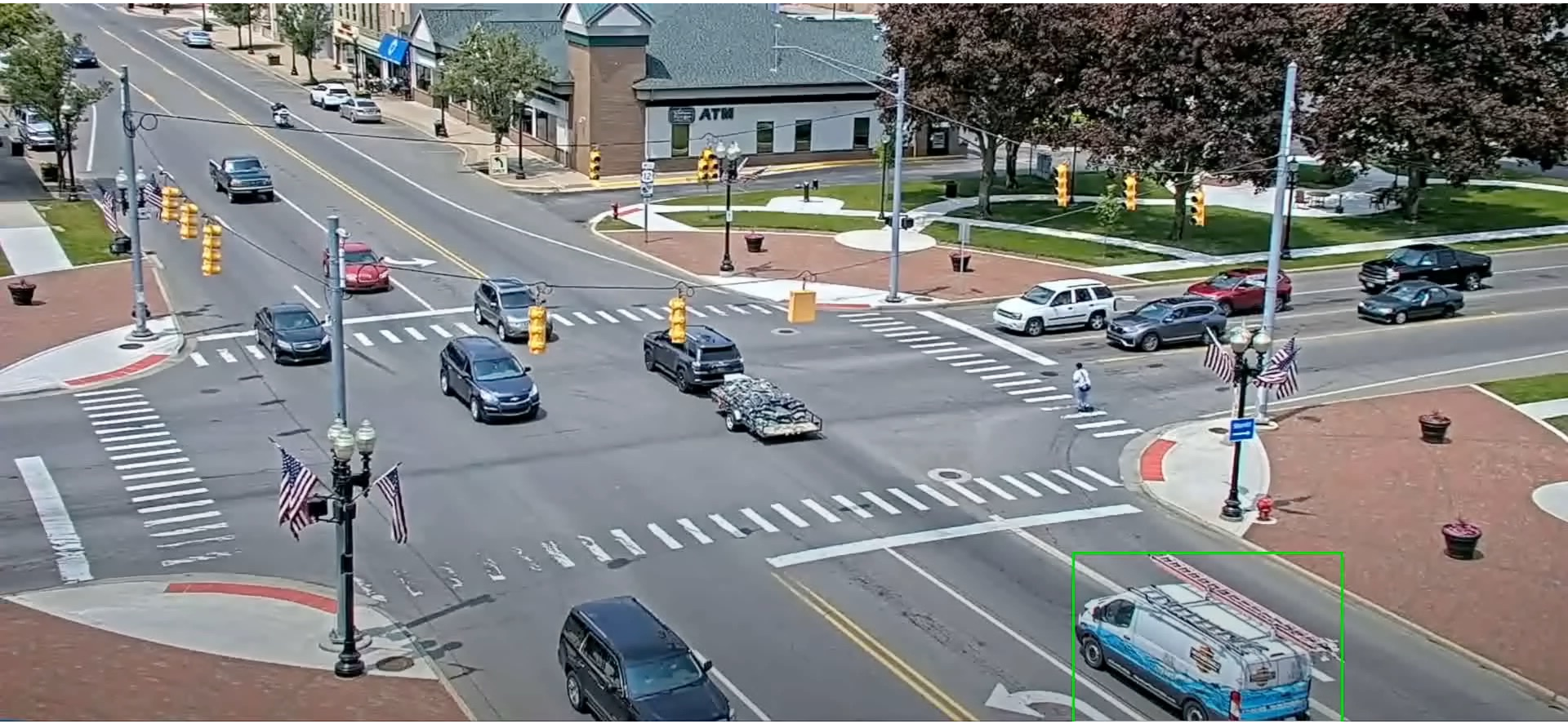
007_2.txt

```
3
3 0
5 1
7 0
```

007_3.txt

```
3
2 1
5 1
7 0
```

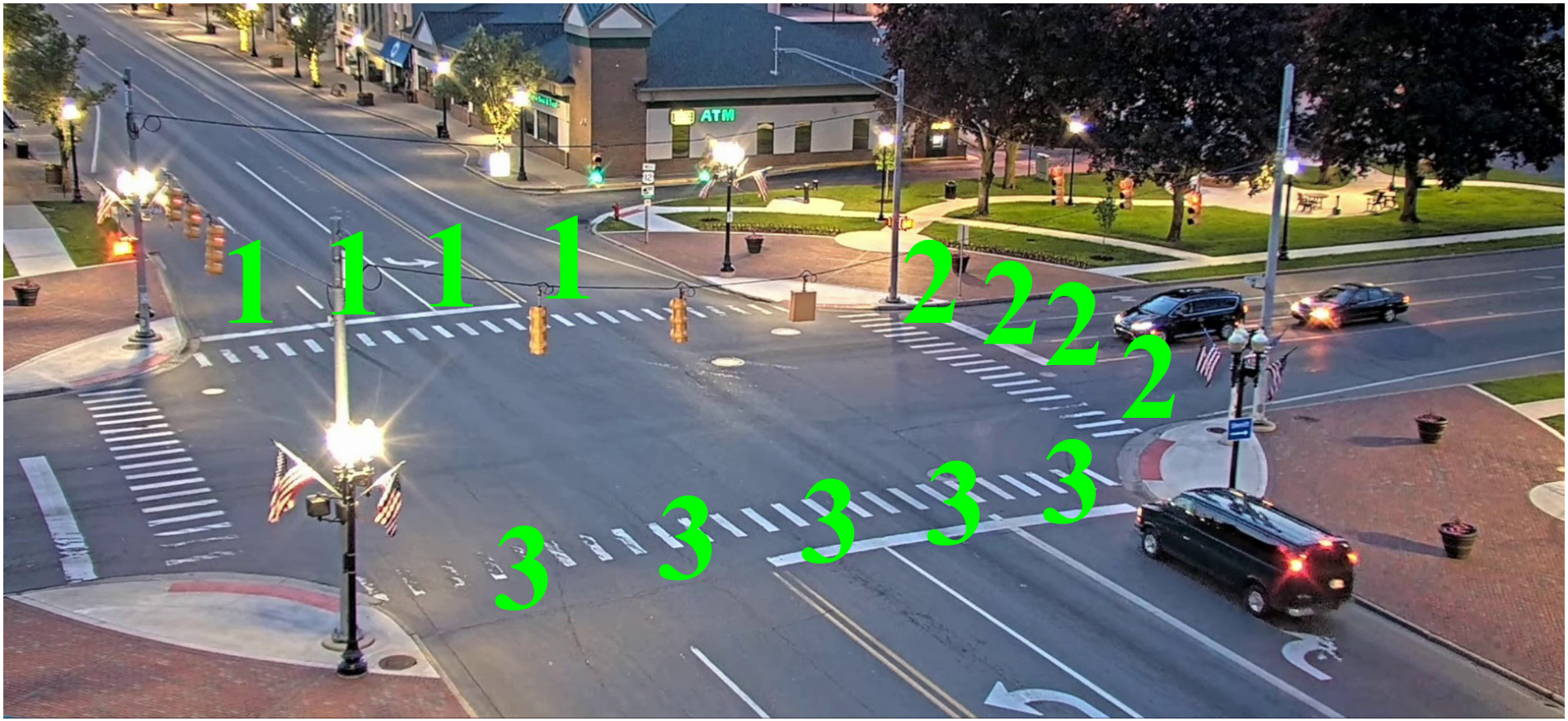
Task 2 – track a specific vehicle



- the task is to track a specific vehicle in a given video (you are given the initial bounding box in the first frame of the video)

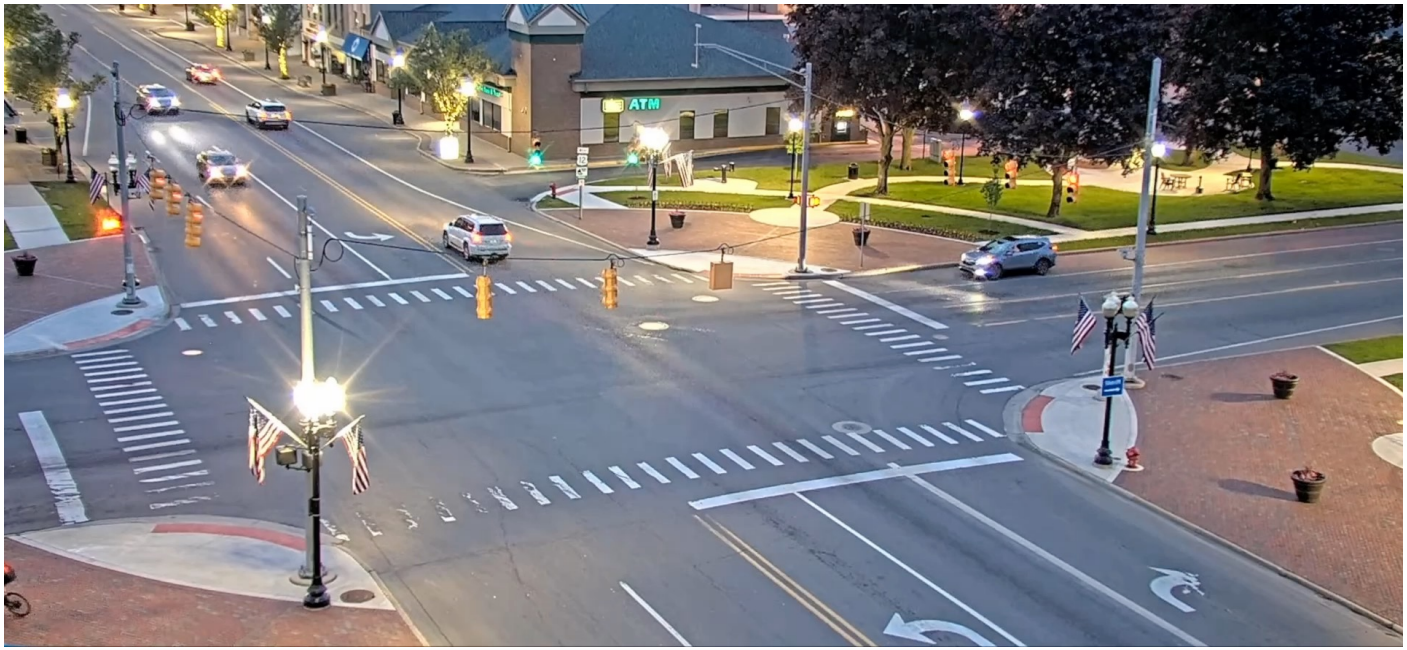
Task 3 - Region numbering

3. Count the trajectories in a video



Task 3 – count the trajectories

3. Count the trajectories in a video



1-2: 1
1-3: 3
2-1: 2
2-3: 2
3-1: 0
3-2: 2