Project Guide

Main steps

- Form a group (min. 3, max. 6 people) and select a topic
    - Graduate students may do individually
- Introduce the problem and difficulties
- Short survey about the problem, common approaches
- Find or collect a dataset (face images, DNA sequences etc...)
- Select one (or more) suitable method to utilize (Fourier transform, region-based histogram equalization, GMM, HMM, kNN, SVM, ANN, PCA, RBM, CNN, RNN etc...)
- Experiment on the dataset, using any platform of choice (Matlab, Python, Java, Kotlin, C#, C++, C, Qbasic, HTML, Jquery, R, Assembly or any other)
- **<u>Write everything in your own words and understanding (don't copy-paste, give the reference (citation) instead of copying)</u>**
- Check the originality of your report in **TurnItIn** with the corresponding assignment (e.g. Final)
    - Class ID: **38725510**
    - Key: **image**
- Submit your project reports to MS Teams (only one member to submit)
    - Attach your full source code (project) in the submission
    - Name your zip file as group_id.zip, e.g. 12.zip
    - Attach the **TurnItIn** originality check report (find the two distinct assignments for intermediate and final reports)
- Group IDs to be announced

Grading

For **task-oriented projects**

(Majority of the projects, similar to a scientfic research, e.g. For face detection)

Unless explicitly stated and agreed by me, your project is task-oriented by default

Have a good understanding of the problem and investigate different solutions

- **Project intermediate report**
- 40% of project grade (16 points overall the course)
- Introduction & problem & literature survey (50% of report grade)
- Describe the methods to use and compare (50% of report grade)

- **Project final report**
- 60% of project grade (24 points overall the course)
- Introduction & problem & literature survey (10% of report grade)
- Describe the methods to use and compare (25% of report grade)
- Describe the dataset (6% of report grade)
- Show your experimental results (34% of report grade)
- Attach your source code (part that you write on your own) to the report (25% of report grade)

Grading

For **implementation-oriented** projects, e.g. implement CNN from scratch, implement level-set from scratch for segmentation etc…

- **Project intermediate report**
- 40% of project grade (16 points overall the course)
- Introduction & problem & survey (20% of report grade)
- Describe the method (40% of report grade)
- Describe the dataset (10% of report grade)
- Attach the completed portion of the source code that you write (30% of report grade)

- **Project final report**
- 60% of project grade (24 points overall the course)
- Introduction & problem & survey (7% of report grade)
- Describe the method (20% of report grade)
- Describe the dataset (5% of report grade)
- Show your experimental results (33% of report grade)
- Attach your source code (part that you write on your own) to the report (35% of report grade)

Deadlines

Project intermediate report **(Due 2023.05.25)**

Project final report **(Due 2023.06.18)**

Late submission policy
First 6 hours: 10% discount on the (report) grade
After first 6 hours to 24 hours: 20% discount on the (report) grade
More than 24 hours: 100% discount on the (report) grade

Turnitin policy

- No discount up to 30% similarity
- (similarity / 2) points discount for more than 30% and less than 75% similarity
    - (e.g. 25 points discount for 50% similarity)
- No grades for more than 75% similarity

Details about the steps

Example topics

If you don’t have an idea, you can select one of those topics

Both kind of projects (task or implementation) will be OK for any topic but the suggested project kind is shown in parantheses:

- ❖ Level set implemantation (implementation-oriented)
- ❖ Semantic segmentation (task-oriented)
- ❖ Face detection / face recognition / general object detection (task-oriented)
- ❖ Counting the value of coins in a scene (implementation-oriented **only**)
- ❖ Implementing CNN (implementation-oriented)
- ❖ Object tracking in video (task-oriented)
- ❖ Alignment / estimation of transform or deformation (both kind of projects will be **OK**)
- ❖ Camera calibration (both kind of projects will be **OK**)
- ❖ Medical image processing (e.g. finding a disease in a CT image)
- ❖ Self-driving car (task-oriented)
- ❖ Image Super-Resolution (task-oriented)
- ❖ Image Synthesis (task-oriented)
- ❖ Remote sensing (task-oriented)
- ❖ Text to Image (task-oriented)

Introduce the problem

- Provide a brief introduction about the problem
  - What is the problem (e.g. recognizing a subject from the face image)
  - What are the difficulties (e.g. pose and illumination variabilities...)
  - ...

Survey

Related or similar works in the literature

Find common approaches used in the context, explain each method in one paragraph

Provide your references at the end

If your project is the first for some task (e.g. Recognizing alien gender from distance)
    You can still describe and reference similar methods
    (e.g. Previous works for human gender classification etc…)

Method

- Explain in detail the method (or methods in task-oriented projects) that you use
- Explain how you utilize the method, e.g.
  - We preprocess the image and extract ... features
  - ...

- This should be the most detailed section

Dataset

- Briefly explain the dataset(s) that you use
- Add the references (article or web site etc…)
- Give information, e.g.
    - It consists of 100 subjects
    - Each subject provides 5 face images
    - Images are taken in a single session
- ...

Experimentation

- Explain the experimental protocol in detail, e.g.
  - We use the first 3 face images of each subject for training the model
  - We test the model using the remaining 2 face images
- Provide your results (both qualitative and quantitative) in detail, e.g.
  - We achieved 90% accuracy using ... method
  - We achieved 95% accuracy when we use 4 face images
  - Accuracy dropped to 80% when there exist noise in the image
  - 70% precision and 90% recall in the segmentation of face region
  - Result of a bad segmentation is shown in Figure 1
  - ...

Source Code

For task-oriented projects:

- Include important parts that you write on your own to the report
- You can use libraries or existing codes when necessary
  - Example: You are doing object recognition
  - You may use libraries for different object recognition algorithms
  - Run them on datasets and compare their performances

For implementation-oriented projects:

- Include all codes that you write
- You can use libraries for low-level operations depending on the topic; such as libraries for matrix operations etc...

Codes should be clearly commented
The parts you write on your own should be indicated

Examples of steps

These are just small parts to provide some intuition
Your actual report should be much comprehensive and detailed

All steps should be taken for implementation-oriented and task-oriented projects (for all projects)

<u>For example:</u> Your project is to **implement a convolutional neural network (CNN) for face recognition**

You should give an introduction on CNNs and face recognition

You should both describe important works about CNNs (works that introduce and improve CNNs) and also face recognition

You should describe and formalize your CNN model and how you recognize faces with your model, in Method section

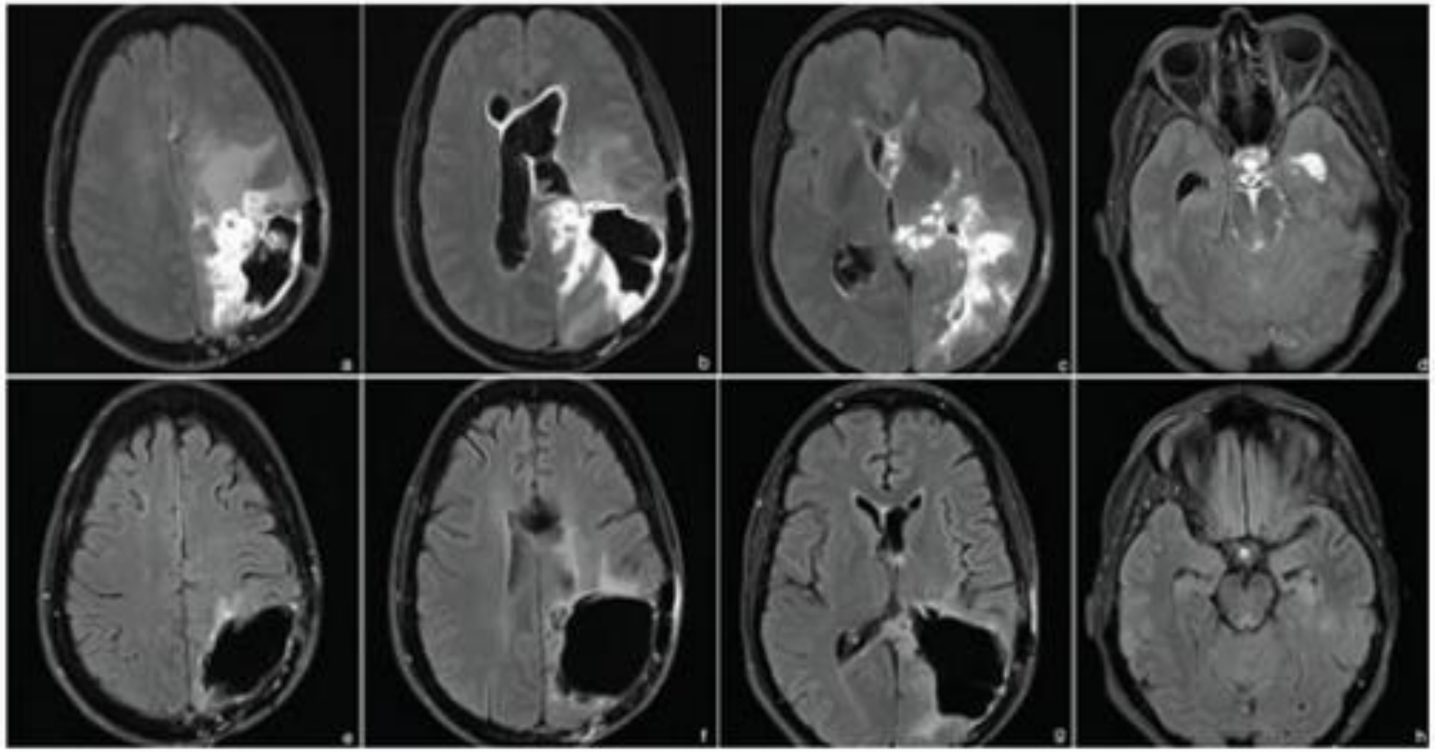
You should test your implementation on some data (e.g. Face data)

You should describe the face dataset you use for training and testing

You should provide the results (e.g. Accuracy) you get with your implementation

Introduce the problem

Glioblastoma is the most common high grade (cancerous) primary brain tumor in adults. They can also occur, rarely, in children. Glioblastomas belong to a group of brain tumors known as gliomas, as they grow from a type of brain cell called a glial cell. The current standard of treatment is maximal surgical resection, followed by radiation, with concurrent and adjuvant chemotherapy.



Due to the heterogenerity most tumors develop resistance to treatment and shortly recur. Tumors evolve over time to boost their DNA repair in an effort to stave off the damage inflicted with radiation or chemotherapy. Following recurrence glioblastoma is quickly fatal in the majority of cases.

With the aid of powerful and accessible image editing tools, image post-processing and refinement have become more and more popular for many people. In some cases, the post-processed images to create fraudulent contents that are occurred in court evidence, news report, commercial contract, and so on, has caused serious social problems [1]. Hence, the image credibility authentication has become a major concern in image forensics and security. The credibility authentication also means detecting the forgery image to find the tampered snippets. Copy move forgery is one of the most actively subtopic in forgery scheme [2]. A general method of Copy Move Forgery Detection (CMFD) is divided into two independent

Survey

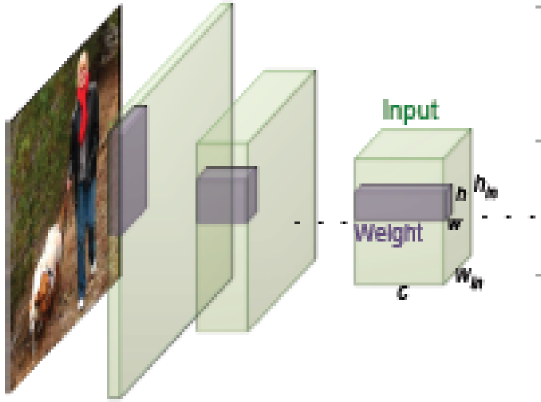
are considered in different works. Tolosana et al. analyze the feasibility of RNNs for on-line signature verification in real practical scenarios [8]. RNN (with LSTM layer) outperformed the previous results on the BiosecurID [9] ranging from 17.76% to 28.00% improvement for skilled forgeries.

CNNs are combined with RNNs usually in video processing tasks. Such hybrid networks are considered frequently also in offline handwriting recognition problem. Hybrid CNN-RNN architecture has been utilized for recognizing Urdu text from printed documents [10]. The network transcribes a sequence of convo-

References

1. Alonso-Fernandez, F., Fairhurst, M.C., Fierrez, J., Ortega-Garcia, J.: Impact of signature legibility and signature type in off-line signature verification. In: 2007 Biometrics Symposium, pp. 1–6 (2007)
2. Galbally, J., Gomez-Barrero, M., Ross, A.: Accuracy evaluation of handwritten signature verification: Rethinking the random-skilled forgeries dichotomy. In: 2017 IEEE International Joint Conference on Biometrics (IJCB), pp. 302–310 (2017)
3. Graves, A., Schmidhuber, J.: Offline handwriting recognition with multidimensional recurrent neural networks. In: D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (eds.) Advances in Neural Information Processing Systems 21, pp. 545–552. Curran Associates, Inc. (2009)
4. Byeon, W., Breuel, T.M., Raue, F., Liwicki, M.: Scene labeling with LSTM recurrent neural networks. In: CVPR, pp. 3547–3555. IEEE Computer Society (2015)

Method



	Network Variations	Operations used in Convolution	Memory Saving (Inference)	Computation Saving (Inference)	Accuracy on ImageNet (AlexNet)
Standard Convolution	Real-Value Inputs 0.11 -0.21 ... -0.34 -0.25 0.61 ... 0.52; Real-Value Weights	+ , − , x	1x	1x	%56.7
Binary Weight	Real-Value Inputs 0.11 -0.21 ... -0.34 -0.25 0.61 ... 0.52; Binary Weights	+ , −	~32x	~2x	%56.8
BinaryWeight Binary Input (XNOR-Net)	Binary Inputs 1 -1 ... -1 -1 1 ... 1; Binary Weights	XNOR , bitcount	~32x	~58x	%44.2

Fig. 1: We propose two efficient variations of convolutional neural networks. **Binary-Weight-Networks**, when the weight filters contains binary values. **XNOR-Networks**, when both weigh and input have binary values. These networks are very efficient in terms of memory and computation, while being very accurate in natural image classification. This offers the possibility of using accurate vision techniques in portable devices with limited resources.

In this paper, we introduce simple, efficient, and accurate approximations to CNNs by binarizing the weights and even the intermediate representations in convolutional neural networks. Our binarization method aims at finding the best approximations of the convolutions using binary operations. We demonstrate that our way of binarizing neural networks results in ImageNet classification accuracy numbers that are comparable to standard full precision networks while requiring a significantly less memory and fewer

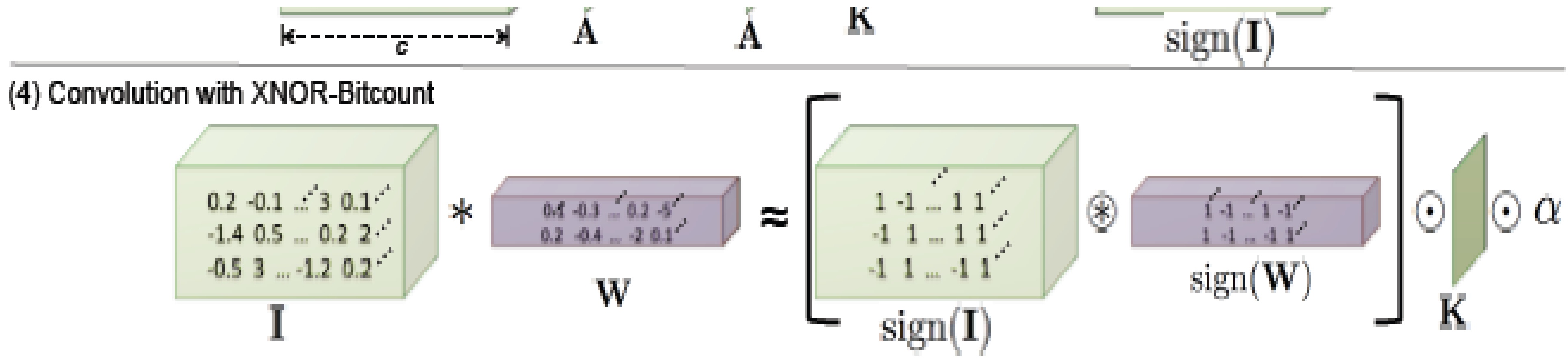


Fig. 2: This figure illustrates the procedure explained in section 3.2 for approximating a convolution using binary operations.

the optimal solutions can be achieved from equation 2 as follow

$$\mathbf{C}^* = \text{sign}(\mathbf{Y}) = \text{sign}(\mathbf{X}) \odot \text{sign}(\mathbf{W}) = \mathbf{H}^* \odot \mathbf{B}^* \tag{9}$$

Since $|\mathbf{X}_i|, |\mathbf{W}_i|$ are independent, knowing that $\mathbf{Y}_i = \mathbf{X}_i \mathbf{W}_i$ then, $\mathbf{E}\left[|\mathbf{Y}_i|\right] = \mathbf{E}\left[|\mathbf{X}_i||\mathbf{W}_i|\right] = \mathbf{E}\left[|\mathbf{X}_i|\right] \mathbf{E}\left[|\mathbf{W}_i|\right]$ therefore,

$$\gamma^* = \frac{\sum |\mathbf{Y}_i|}{n} = \frac{\sum |\mathbf{X}_i||\mathbf{W}_i|}{n} \approx \left(\frac{1}{n}\|\mathbf{X}\|_{\ell 1}\right)\left(\frac{1}{n}\|\mathbf{W}\|_{\ell 1}\right) = \beta^* \alpha^* \tag{10}$$

Dataset

Recognition Algorithm for GTSRB Dataset

The German Traffic Sign Recognition Benchmark (GTSRB) was a multi-class, single-image classification challenge held at the International Joint Conference on Neural Networks (IJCNN) 2011, with the following requirements:

- 51,840 images of German road signs in 43 classes (Figures 14 and 15)
- Size of images varies from 15x15 to 222x193
- Images are grouped by class and track with at least 30 images per track
- Images are available as color images (RGB), HOG features, Haar features, and color histograms
- Competition is only for the classification algorithm; algorithm to find region of interest in the frame is not required
- Temporal information of the test sequences is not shared, so temporal dimension cannot be used in the classification algorithm

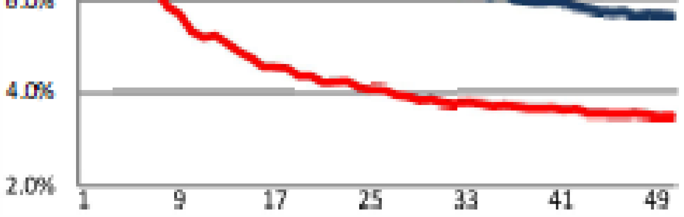
4.1 Databases

GPDS-960 and GPDS-Synthetic-10000 signature databases are utilized to evaluate the proposed techniques and confirm the benefits. GPDS-960 signature database consists of 881 users. Gray-level signature images are in png format and have been scanned at 600 dpi. Each subject has 24 genuine samples and at most 30 forgery samples. Number of forgery samples is less than 30 for a few subjects.

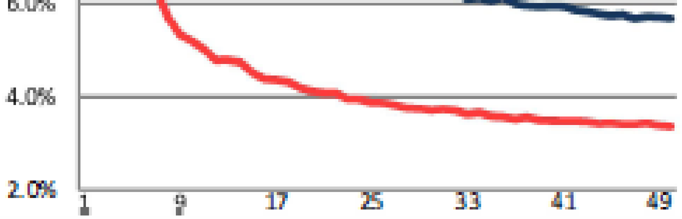
GPDS-Synthetic-10000 online and offline bimodal database [31] is a challenging synthetic database consisting of 24 genuine and 30 skilled forgery signatures for each individual. All static signatures are generated with different modeled pens. We only use the static signature images and ignore the dynamic data. This database is the most recent large-scale synthetic signature database and it is not to be confused with the previous GPDS-Synthetic-4000 offline signature

The databases we used are the CASIA-OLHWDB1.0 (DB1.0) and CASIA-OLHWDB1.1 (DB1.1) [4], which were collected by the Institute of Automation of the Chinese Academy of Sciences. DB1.0 contains 3740 characters of GB1 from 420 writers, 336 allocated for training and 84 for testing, and DB1.1 contains 3755 classes of GB1 from 300 writers, 240 for training and 60 for testing.

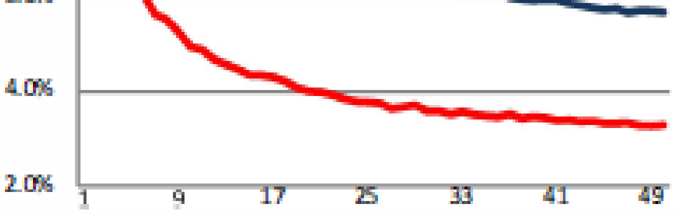
Results



(d) 8-direction Features



(e) Imaginary Strokes



(f) Fusion with four domain methods

Fig. 4. The performance in error rate of different methods on CASIA-OLHWDB1.0. The x-axis indicates the number ($\times 10^4$) of mini-batch.

TABLE I. RECOGNITION RATES (%) OF DOMAIN KNOWLEDGE

Network (3.8 mil.)	*Methods*	*OLHWDB1.0 (38 epoches)*	*OLHWDB1.1 (62 epoches)*
A	Sign0 (**Baseline**)	94.32	93.98
B	Baseline+Sign1	96.24	95.87
C	Baseline+Sign2	96.41	96.10
D	Baseline+Sign2+DT	96.48	96.12
E	Baseline+Sign2+NLN	95.69	95.89
F	Baseline+Sign2+8Dir	96.50	96.18
G	Baseline+Sign2+IS	96.64	96.31
H	Baseline+Sign2 +DT+8Dir+IS	**96.72**	**96.35**

conduct the following experiments incorporated with Sign2 feature maps.

We performed five more experiments (denoted by D through E in Table I) to intensively evaluate the effects produced by the

TABLE II. RECOGNITION RATES (%) OF DCNNS ENSEMBLING

Database	*Single Network*		*DCNNs Ensembling (A~H)*		
	Baseline	*Fusion (proposed)*	*Voting*	*Averaging*	*HSP-DCNN (proposed)*
DB1.0	94.32	96.72	97.16	97.19	**97.20**
DB1.1	93.98	96.35	96.84	96.86	**96.87**

DT serves to extend it, especially when data are insufficien Second, the NLN method in [5] (Fig.4c) produces better result than other NLN methods in our preliminary experiments wit small categories, but being of inferior performance than networ C as shown in Table I, consistent with the limitation noted i Section III, regardless of whether the NLN was applied befor or after the ED. Third, with the use of 8 directional feature map (8Dir) with results as shown in Fig.4d, even though the Sign feature maps contain six maps of the first and second iterated integrals signature, which already include the directiona information, the additional 8 directional features ar complementary to them, resulting from the statistical effect o directions, where the traditional CNN is helpless. Furthermore providing imaginary strokes (IS) to a character, resulting i when $N = 5$ and $N = 12$ accordingly. In each table, EER results with both global and user-based thresholds are provided. Among WD models, RBP utilizes RNN and others utilize SVM classifiers. As a reminder, experiments are repeated for random reference and test sets as described in Section 4.2; mean and standard deviation of such experiments' results are shown in the tables.

Table 2: GPDS-160 WD classifier EER results for $N = 5$.

Method	Global th.	User-based th.
RBP size 300 (no hist. select.)	$6.75 \pm 0.22\%$	$5.16 \pm 0.19\%$
(1) RBP size 220 (proposed hist. select.)	$5.77 \pm 0.13\%$	$4.29 \pm 0.15\%$
(2) Proposed CNN	$7.34 \pm 0.45\%$	$4.34 \pm 0.12\%$
(3) SigNet-F [21]	$4.28 \pm 0.23\%$	$2.83 \pm 0.29\%$
(4) SigNet-SPP [23]	$4.86 \pm 0.32\%$	$3.44 \pm 0.27\%$
Combi. 1 & 2	$3.38 \pm 0.28\%$	$1.82 \pm 0.14\%$
Combi. 1 & 3	$2.41 \pm 0.28\%$	$1.34 \pm 0.10\%$
Combi. 2 & 3	$2.97 \pm 0.11\%$	$1.87 \pm 0.12\%$

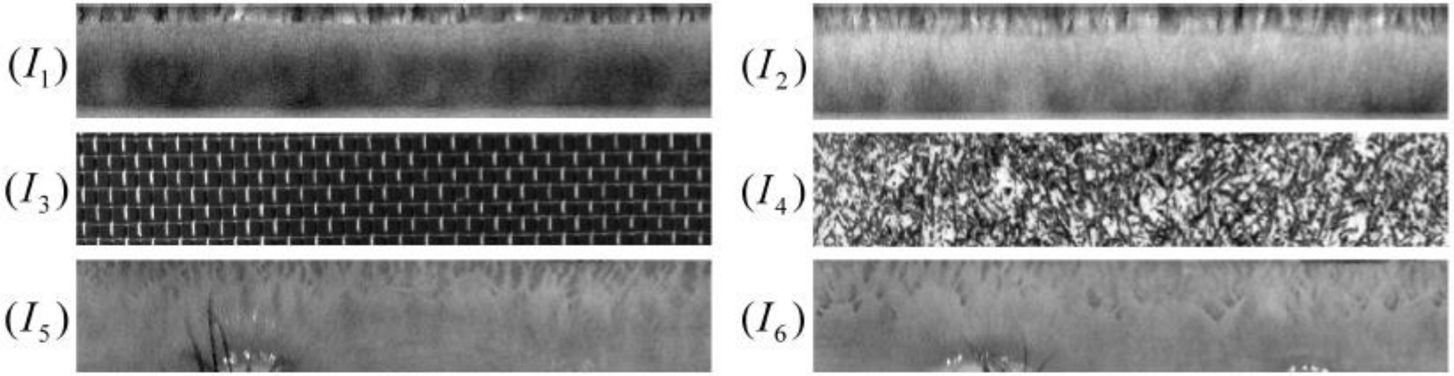
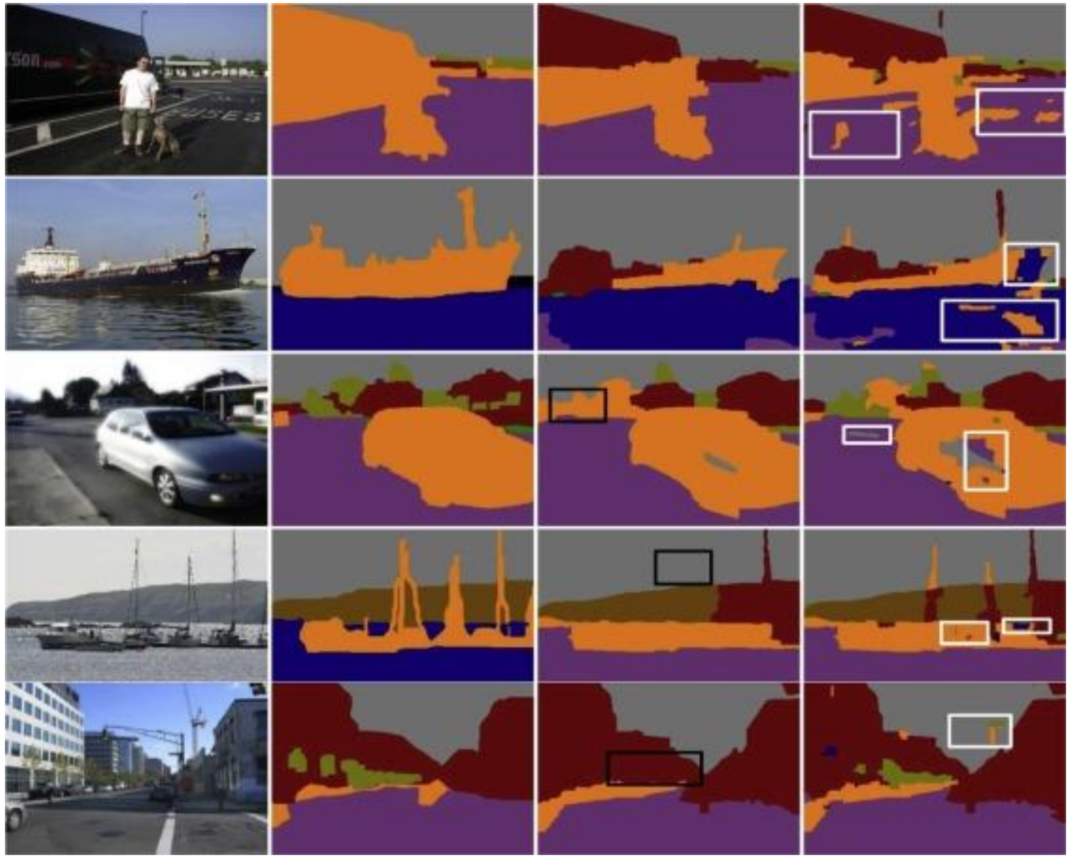
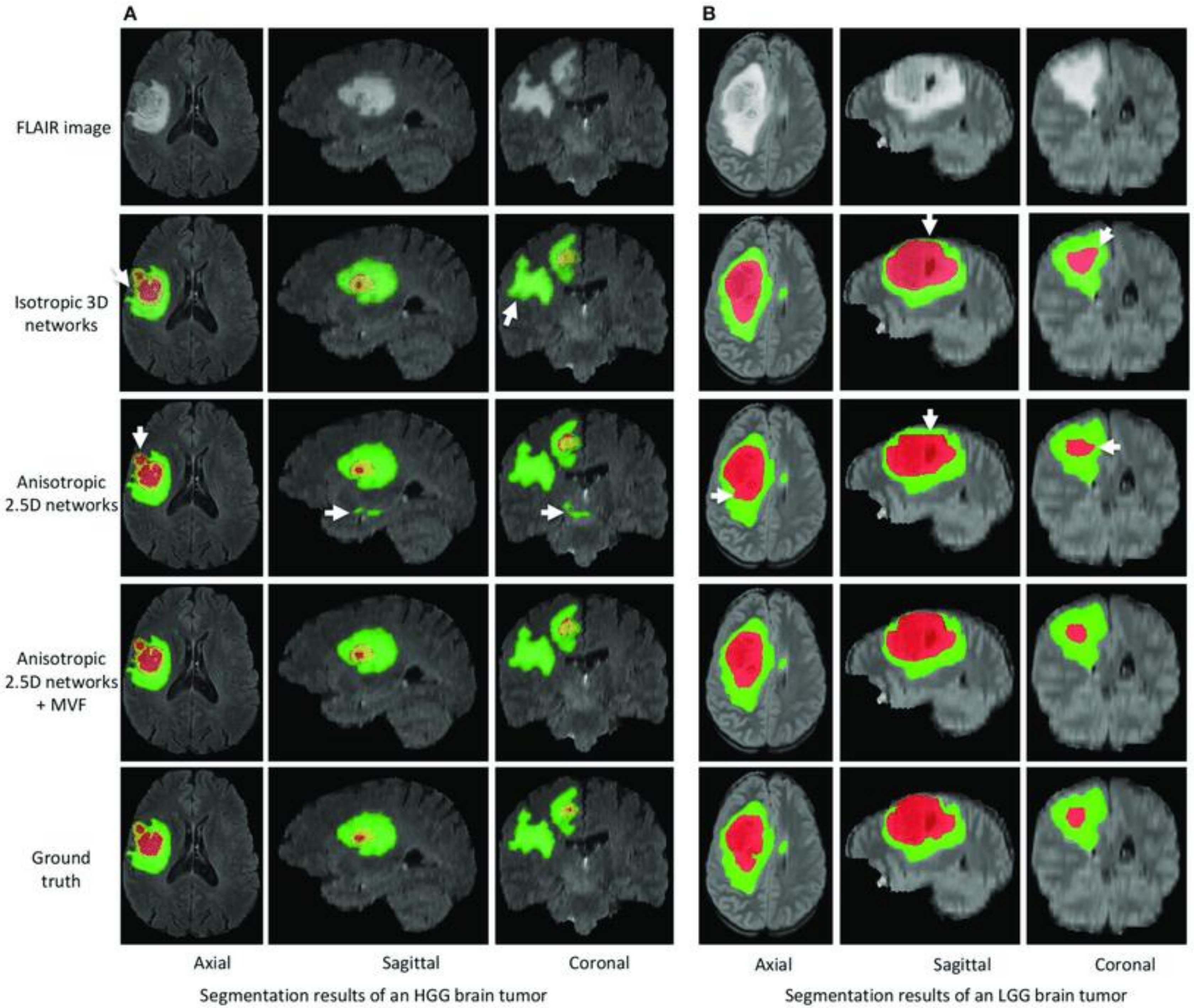


Fig. 16. Original images. I_1 and I_2 are synthetic iris images, I_3 and I_4 are natural images and I_5 and I_6 are real iris images.




A
B
FLAIR image
Isotropic 3D networks
Anisotropic 2.5D networks
Anisotropic 2.5D networks + MVF
Ground truth
Axial
Sagittal
Coronal
Segmentation results of an HGG brain tumor
Segmentation results of an LGG brain tumor