
**VILNIUS UNIVERSITY**

**FACULTY OF MATHEMATICS AND INFORMATICS**

**INFORMATICS STUDY PROGRAMME**

Master's thesis

# Analysis of Computed Tomography Images of the Pancreas to Identify the Site of Cancer

## Kasos kompiuterinės tomografijos vaizdų analizė siekiant nustatyti vėžio pažeistą vietą

Jelizaveta Lemeševa

Supervisor : prof. habil. dr. Gintautas Dzemyda

Reviewer : dr. Andrius Vytautas Misiukas Misiūnas

**Vilnius**
**2025**

## Summary

Pancreatic cancer is a lethal disease that is hard to detect in the early stages. Poor outcomes of pancreatic cancer make the scientific community look for ways to improve cancer detection time and potentially save patients' lives. Automating tumor detection is one of the ways to achieve this. While radiological image classification can make the process cheaper and faster, image segmentation is important for detection of tumor edges, which aids diagnosis and treatment. One of the ways to detect pancreatic tumors in radiological images is deep learning. U-Net is baseline standard for medical image segmentation and so far, performs very well on most tasks, though not so well on pancreatic cancer segmentation. Another way of research would be to try various computer vision algorithms to detect keypoints in pancreatic cancer computed tomography (CT) images, which can be used in conjunction with various machine learning algorithms. The ways to do it could be using computer vision methods like *Scale Invariant Feature Transform (SIFT)*, *Speeded Up Robust Feature (SURF)*, *Features from Accelerated Segment Test (FAST)*, *Oriented FAST and Rotated BRIEF (ORB)*, *Binary Robust invariant scalable keypoints (BRISK)*, *KAZE*, *Accelerated KAZE (AKAZE)* or *Adaptive and Generic Corner Detection Based on the Accelerated Segment Test (AGAST)*. Although there is some research on the usage of these methods for detecting keypoints in medical images, it is not extensive and none of it was applied to pancreatic cancer. This work attempts to try out computer vision algorithms which can find the keypoints corresponding to tumor in pancreatic cancer CT images and apply them to the images of Medical Segmentation Decathlon dataset.



**Keywords:** Computed Tomography, computer vision, OpenCV, pancreatic cancer

# Santrauka

Kasos vėžys yra pavojinga ir mirtina liga, kurią sunku nustatyti ankstyvoje stadijoje. Dėl prastos kasos vėžio prognozės ieškoma būdų, kaip pagreitinti vėžio aptikimo procesą. Vienas iš būdų tai padaryti yra naviko aptikimo automatizavimas. Radiologinių vaizdų automatinis klasifikavimas sutrumpina diagnozavimo laiką, o vaizdo segmentavimas yra svarbus naviko kraštams nustatyti. Vienas iš būdų aptikti kasos navikus radiologiniuose vaizduose yra gilusis mokymasis (angl. deep learning). Dirbtinis neuroninis tinklas „U-Net“ yra medicininių vaizdų segmentavimo standartas ir labai gerai atlieka daugumą segmentavimo užduočių, tačiau ne taip gerai veikia su kasos vėžio vaizdais. Kita mažai tyrinėta kryptis yra kompiuterinės regos metodų, skirtų kontrolinių taškų aptikimui vaizduose, taikymas kasos vėžio vaizdams. Šiems metodams priklauso *Scale Invariant Feature Transform (SIFT)*, *Speeded Up Robust Feature (SURF)*, *Features from Accelerated Segment Test (FAST)*, *Oriented FAST and Rotated BRIEF (ORB)*, *Binary Robust invariant scalable keypoints (BRISK)*, *KAZE*, *Accelerated KAZE (AKAZE)* arba *Adaptive and Generic Corner Detection Based on the Accelerated Segment Test (AGAST)* ir kiti, ir jie gali būti naudojami kartu su mašininio mokymosi metodais. Nors yra tam tikrų šių metodų taikymo medicininiams vaizdams tyrimų, jų nėra daug, ir nėra tyrimų specifiškai apie kasos vėžį. Šiame darbe siekiama ištirti aukščiau minėtų kompiuterinės regos algoritmų galimybes aptikti kontrolinius taškus, atitinkančius naviką, kasos vėžio kompiuterinės tomografijos vaizduose, paimtuose iš *Medical Segmentation Decathlon 2018* duomenų bazės.

**Raktiniai žodžiai:** Kompiuterinė tomografija, kompiuterinė rega, OpenCV, kasos vėžys

# Contents

# Introduction

Pancreatic cancer is a lethal disease that is hard to detect in the early stages. Because of its poor survival rate, the scientific community looks for ways to improve cancer detection time and improve the outcomes. Automating tumor and its edges' detection is one of the ways to do this.

Computed Tomography (CT) is a noninvasive imaging method of patients' internal organs. It visualizes tissues inside of the body which cannot be seen with plain X-ray, like tumors, which makes it one of the most used methods for pancreatic cancer detection. The first practical CT instrument was developed in the 1970s by Godfrey N. Hounsfield for which he received the Nobel Prize in 1979. Since then, CT technology has developed a lot and became a standard imaging procedure for treatment planning, diagnostic, therapeutic, interventional purposes, etc. [Liu18] CT is commonly used for imaging of pancreatic tumors.

**Relevance.** Pancreatic cancer was the 12th most common cancer in 2020 with 495 000 new cases worldwide. GLOBOCAN cancer statistics report estimates that pancreatic cancer rates will increase by 70% in 20 years, with around 844 000 new cases each year. More than half of newly diagnosed cases are metastatic, and the 5-year survival rate for pancreatic cancer in Nordic European countries is around 10%. This poor survival rate is due to factors such as non-specific symptoms and anatomical location, and because the pancreas is close to main vessels, less than 20% of tumors are operable when diagnosed. If treatment is still possible, surgery has mortality risk around 1%, which is considered high.

In Lithuania, 8514 pancreatic cancer cases (men - 4364 cases, women - 3150 cases) were diagnosed and 7684 persons died from pancreatic cancer in the period of 1998–2015. Pancreatic cancer rates were higher for men, and the rates of it only increased in the mentioned period, especially in the 50–74 years age group. Main risk factors of pancreatic cancer are smoking, obesity, lack of physical activity, poor diet, and diabetes, and they are quite high in Lithuanian population, which increases prevalence of pancreatic cancer. [KDK+22].

Poor outcomes of pancreatic cancer make the scientific community look for ways to improve cancer detection time and potentially save patients' lives. Automating tumor detection is one of the ways to achieve this. While radiological image classification can make the process cheaper and faster, image segmentation is important for detection of tumor edges, which aids diagnosis and treatment. Chemoradiation therapy is a treatment for borderline resectable and locally-advanced pancreatic ductal adenocarcinoma, the most prevalent form of pancreatic cancer. It is necessary to assess tumor volume before applying this treatment. The assessment is time consuming and complex and would greatly benefit from automation [MKR+22].

Various machine learning and deep learning methods are being used to detect pancreatic tumors in radiological images. However, pancreas itself varies greatly in its size and shape in different patients, and it is even harder to detect cancerous regions in it, hence there is space for improvement in pancreatic cancer detection [PAK+21].

**Novelty.** Most pancreatic cancer detection research is focused on finding deep learning neural networks architectures that would perform better than U-Net. U-Net is baseline standard for medical

image segmentation and so far, performs very well on most tasks, though not so well on pancreatic cancer segmentation. The best achieved result is by Zhang et al. who used nnUNet network on a large dataset of multi-phase CT images of pancreatic ductal adenocarcinoma, which achieved Dice scores of 0.709 ± 0.159 and 0.522 ± 0.250 for multi and venous phase respectively. Other researchers have achieved Dice scores between 0.54 and 0.64 [MKR+22]. For comparison, brain tumor segmentation mean Dice scores reach 0.865 (though it uses MRI instead of CT) [ASK+20], and for liver the score is 0.78 [KJL+21]. A similar tendency can also be seen in Medical Segmentation Decathlon leaderboard[1] with Dice scores being lower for pancreatic cancer segmentation.

Another research direction is usage of keypoint detectors and descriptors for cancer detection in medical images. The ways to do it could be using methods like Scale Invariant Feature Transform (SIFT), Speeded Up Robust Feature (SURF), Features from Accelerated Segment Test (FAST), Oriented FAST and Rotated BRIEF (ORB) and similar, usually in combination with machine learning methods. For example, ORB patented alternative, Scale Invariant Feature Transform (SIFT), has been used for feature extraction in MRI images to detect brain cancer [AK20] and prostate cancer [CKS20]. Although there is some research on the usage of these methods for tumor detection, it is not extensive and none of it was applied to pancreatic cancer.

**Aim, objectives and expected results.** The aim of this thesis is to find keypoints corresponding to tumor in the slices of 3D Computed Tomography pancreatic cancer images.

Objectives:

1. Review keypoint detection and description algorithms application for medical images.
2. Apply these algorithms for pancreatic cancer CT images from Medical Segmentation Decathlon dataset.
3. Evaluate and compare the algorithms' performance.

The expected result of this Thesis is to investigate keypoint detection and description algorithms which can find the region of cancer in a Pancreatic cancer CT image slice, implement these algorithms or use existing implementations, and apply them to the images, visualize results and provide evaluation metrics. These evaluation metrics should then be compared between different algorithms.

**Data and tools.** 3D CT image consists of 2D images called slices. Slice thickness is determined by the operator of CT scanner and its values are usually between 1 and 10 mm. The thinner the slice, the more image noise is visible in it, but despite this, often the visibility of a tumor is improved by providing more visual content. Therefore, a balance between slice thickness and image noise is usually sought [AK17].

Medical Segmentation Decathlon is a large, annotated dataset of medical images, that was crowd-sourced and made available in 2018. It consists of 2633 3D images, 420 of which are of the pancreas. Images were provided by Memorial Sloan Kettering Cancer Center (New York, NY, USA), the tissue was manually segmented by an expert abdominal radiologist [SAB+19]. The size of the archived pancreas data is around 11 GB, or 15 GB unarchived. CT image format is nifti. The format

[1] https://decathlon-10.grand-challenge.org/evaluation/challenge/leaderboard/

was developed by the Neuroimaging Informatics Technology Initiative (NIfTI) which works with the user and developer communities to address compatibility between different neuroimaging tools and enable scientists to rigorously compare their findings. The primary goal of NIfTI is ”to provide coordinated and targeted service, training, and research to speed the development and enhance the utility of informatics tools related to neuroimaging”, with an initial focus on tools that are used in fMRI[2].

OpenCV (Open Source Computer Vision Library)[3] includes several hundred of computer vision algorithms. As stated on the mentioned website, OpenCV ”provides a common infrastructure for computer vision applications and accelerates the use of machine perception”. OpenCV has a modular structure and includes several libraries: Core functionality, Image Processing, Video Analysis, Object Detection, and others. OpenCV is written natively in C++, but also has Python, Java and MATLAB interfaces and supports all major operating systems. A full-featured CUDA and OpenCL interfaces are being actively developed.

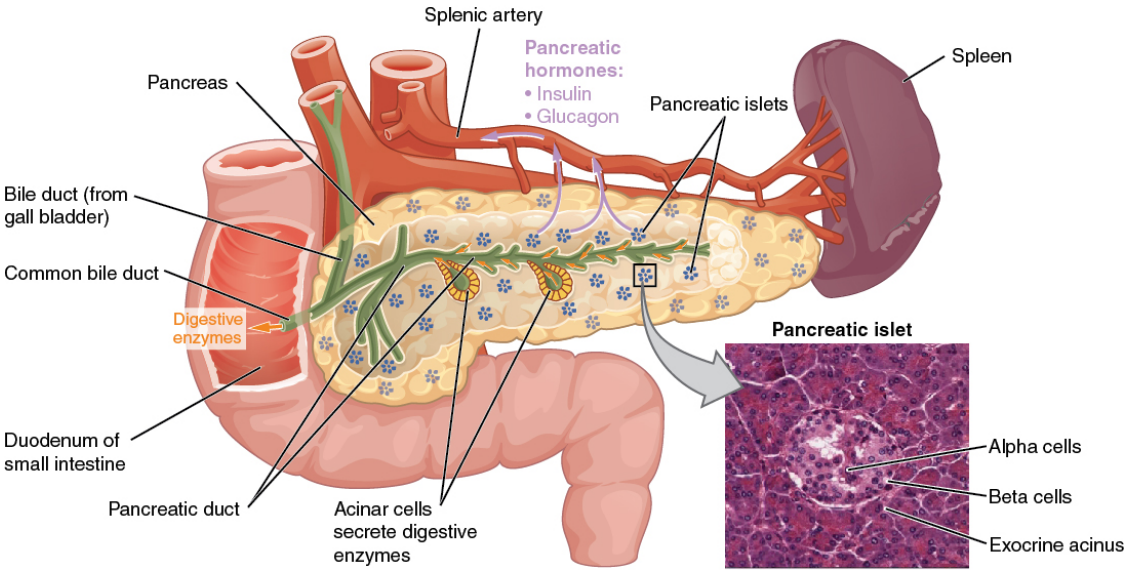
OpenCV has implemented various computer vision algorithms: SIFT[4], SURF[5], FAST[6], and others. It has also implemented ORB[7] as an alternative to SIFT and SURF, which uses less computational resources and is not patented.

[2] https://nifti.nimh.nih.gov/background
[3] https://opencv.org/about/
[4] https://docs.opencv.org/3.4/da/df5/tutorial_py_sift_intro.html
[5] https://docs.opencv.org/3.4/df/dd2/tutorial_py_surf_intro.html
[6] https://docs.opencv.org/3.4/df/d0c/tutorial_py_fast.html
[7] https://docs.opencv.org/3.4/d1/d89/tutorial_py_orb.html

# 1 Pancreatic Cancer and Its Imaging

## 1.1 Pancreas Anatomy and Functions

The pancreas is a long, well-defined, solitary organ, most of which is located posterior to the bottom half of the stomach. It has an exocrine gland that secrets digestive enzymes, and endocrine cells, which secrete the hormones like glucagon, insulin, somatostatin, and pancreatic polypeptide.[HQK+19] It is 14–18 cm long, 2–9 cm wide and 2–3 cm thick, weighing 50–100g. Macroscopically, pancreas can be divided into 3 parts, that have no specific, visible borders - the head, the body, and the tail. The border between the head, which is C-shaped, and the body, which is located under the stomach and crosses multiple arteries and veins, is considered to be the left side of the superior mesenteric artery. The border between the body and the tail, which is near the hilum of the spleen, is the midpoint in between them.[DRS15] The view of the pancreas is provided in Figure 1 figure..




***1 figure.*** *Pancreas [HQK+19]*

Pancreatic islets are clusters of cells formerly known as the islets of Langerhans. Each islet contains cells with distinct functions. The alpha cells produce the hormone glucagon, which release is triggered by low blood glucose levels, and makes up approximately 20% of each islet. The beta cells produce insulin, that is released during elevated blood glucose levels, and make up approximately 75% of each islet. The delta cells account for 4% of the islet cells, and produce somatostatin, the peptide hormone responsible for inhibition of the release of insulin and glucagon, which is also released by intestines, hypothalamus, and stomach. The last type of cells, pancreatic polypeptide cells, account for about 1% and secrete the pancreatic polypeptide hormone, that is responsible for regulating appetite and endocrine, exocrine secretions.[HQK+19]

Pancreatic duct or duct of Wirsung is a elongated thin structure, resembling a tube, that runs through the length of the pancreas, which transports digestive enzymes produced by the pancreas into the small intestine. It begins at the tail and extends towards the head of the pancreas, where it joins with the common bile duct, and forms the hepatopancreatic ampulla, which enters the first part

of the small intestine. The main function of the pancreas duct is to carry pancreatic secretions like digestive enzymes and bicarbonate ions, from the pancreas to the small intestine. It exhibits branching patterns throughout the pancreas, with smaller ducts called intralobular ducts merging to form larger ducts that forms the main pancreatic duct. Sometimes accessory pancreatic ducts are present, which provide an alternative route for pancreatic secretions to reach the duodenum.[HQK+19]

## 1.2 Pancreatic Cancer

### 1.2.1 Causes, Prevalence and Prognosis

Pancreatic cancer is a lethal disease that is associated with poor outcomes. Although it is fairly infrequent and ranked 9-14 by frequency among other cancers (10–15 cases per 100 000 people in the Western world), it has high mortality rates with 5-year survival below 5%. The age of patients is usually elderly with a median of 68 years. Only 6% of cases are early-onset prior to the age of 50 years, though this group has even worse 5-year survival and needs more aggressive treatment.[AHS+22]

4/5 of the patients have advanced disease or distant metastases and are not candidates for upfront pancreatic resection, hence they have no effective treatment options, and another 1/5, which is early-stage patients eligible for surgical resection, have a 5-year survival rate of <31%. Next-generation genome sequencing data has revealed that pancreatic cancer tumors are highly inter- and intratumor heterogeneous and develop resistance to traditional treatments like chemotherapy or radiation therapy, hence it lacks effective therapeutics.[WZY+21] Henderson et al. states that pancreatic cancer ”will become the second leading cause of cancer death by 2030”, and in many Western countries it is already the case.[AHS+22]

Risk factors of pancreatic cancer are genetic properties (familial risk due to susceptibility gene mutations), chronic pancreatitis, pancreatic cysts, diabetes mellitus, age, occupational exposure, obesity or metabolic syndrome, and lifestyle factors including smoking and alcohol abuse. [WZY+21]

As mentioned before, most patients are diagnosed in advanced cancer stages with metastases and a poor prognosis, however, the patients that undergo surgery combined with adjuvant chemotherapy have better survival chances. The transformation of the tumor to an invasive adenocarcinoma is a gradual process, caused by initial driving mutations. Since initial driving mutation, it may take 10 years until the lesion forms and additional 5 years until metastases develop. If patient has no symptoms and a small lesion, and pancreatic cancer is detected during this time, it is potentially curable, but sometimes genetic errors occur simultaneously in multiple places in the pancreas (chromothripsis). This happens in up to 30% of pancreatic cancers, in which case a tumor size of size ≤0.5 cm already develops distant metastases.[AHS+22]

More than half of the patients already have metastases during diagnosis of pancreatic cancer. Surgery can not be performed in these cases, and the treatment they receive is palliative chemoterapy, which is associated with 5 year survival rate of 3%. if pancreatic cancer was detected earlier, that would dramatically increase overall 5 year survival rate as there would be more patients with localized tumors instead of metastases.[AHS+22]

### 1.2.2 Diagnosis and Treatment

Japanese Pancreatic Case Association Criteria has estabiliched criteria for pancreatic cancer stages:

- Stage I: tumor diameter is less than or equal to 2 cm, no vascular invasion and metastasis;
- Stage II: tumor diameter is greater than 2 cm and less than 4 cm, enveloped cancer cell infiltration, no vascular invasion, metastasis;
- Stage III: tumor diameter is greater than 4 cm, with existence of nearby lymphatic metastasis;
- Stage IV: tumor diameter is greater than 4 cm, with evidence of distant lymph node metastasis.

[ZL20]

The most definite way of diagnosing pancreatic cancer is histopathology and/or cytology, although these diagnostic procedures are invasive. Methods to obtain specimens are:

- endoscopic ultrasonography (EUS) or computed tomography (CT) guided biopsy
- ascites cytology
- exploratory biopsy under laparoscopy or open surgery.

A clear pathological diagnosis is needed for all patients, except those undergoing surgical resection, before formulating a treatment plan.[ZL20]

Pancreatic cancer treatment usually includes chemotherapy, radiotherapy, surgery, and palliative care. Furthermore, targeted therapy, immunotherapy, and microbial therapy efficacy is being researched as an addition to traditional treatments. Which treatment to use is determined by the stage of pancreatic cancer. The only way to actually cure pancreatic cancer is surgery. Surgical resection can be divided into resectable, handover resectable, unresectable (local progression), or combined pancreatic cancer with distant metastases. A full-dose chemotherapy prior to surgery has been proved to be effective to achieve local tumor control, as it may eliminate potential metastatic lesions.[ZL20]

Pancreatic cancer resection includes total pancreatectomy, distal pancreatectomy with splenectomy and pancreaticoduodenectomy. Pancreatic cancer that affects arteries is usually considered unsuitable for surgical resection. Also usually the prognosis is worse in patients with para-aortic lymph node (PALN) metastases, though it is not a contraindication for surgery. Another type of surgery performed is laparoscopic surgery, which allows patients to return to their diet earlier and reduce stay in a hospital, but it is more demanding for doctors. Pancreatic cancer surgery has become quite safe with the risk of postoperative mortality dropping to 3% in the past decade.[ZL20]

### 1.2.3 Types

Pancreas has multiple functions and consists of various types of cells. Pancreas cells are usually classified into exocrine and endocrine cells. Exocrine cells are duct and acinar cells that produce var-

ious gastric enzymes needed for digestion. Endocrine cells are mostly β cells that produce hormones like insulin, which is used by the body for glucose metabolism.[DCC+15]

Pancreatic cancer can start in both types of the cells. Adenocarcinoma is the type of cancer that starts from exocrine cells. Pancreatic ductal adenocarcinoma (PDAC) originates in the epithelial cells lining the pancreatic duct, and accounts for nearly 80% to 90% of pancreatic cancer cases. It usually has increased extracellular matrix and growth factor production, that causes tumor tissue growth, and this prevents chemotherapy from working effectively. Another type of exocrine pancreatic cancer is mucinous tumors, that typically arise from the pancreatic ductal epithelium, and accounts for less that 10% of all tumors. Mucinous tumors are usually less invasive than PDAC. Ductal carcinoma has several morphological types like medullary and colloid carcinoma. Squamous and adenosquamous carcinomas should also be recognized as it usually is related to worse prognosis, but acinar cell cancer has a better prognosis.[DCC+15]

Less than 5% of pancreatic cancers originate in endocrine cells, usually islet cells, including β cells and α cells. These tumors, called Pancreatic Neuroendocrine tumors (PNET), that include insulinomas, glucagonomas and gastrinomas, cause pancreas to produce hormones at high levels, which causes noticeable symptoms.[FAL+05]

## 1.3 Computed Tomography of Pancreatic cancer

Computed tomography (CT) refers to a computerized X-ray imaging technique, that was developed in the 1970s by Godfrey N. Hounsfield for which he received the Nobel Prize in 1979. A narrow beam of X-rays is aimed at a specific place of the patient's body and quickly rotated around the circular opening of a structure called a gantry, producing signals that generate cross-sectional images, called slices. These types of images give more information that standard X-ray. After one slice is captured, the motorized bed is moved forward to capture the next area. The image slices are stacked together to produce 3D view of the inspected organs, which allows detecting various abnormalities like injuries, cysts or tumors. The size of tissue captured in one slice can vary from machine to machine, but is usually 1-10 millimeters.[Liu18]

Bones are easily imaged by CT, but soft tissues have varying ability to stop X-rays, hence they can appear blurry and indistinct. Contrast agents based on iodine are highly visible in X-ray and are injected intravenously during the imaging procedure, which greatly enhances CT image quality. For imaging of digestive system barium based compounds can be administered orally. These substances are considered safe, though they can cause allergic reactions or temporary kidney failure in some predisposed patients.[Liu18]

Imaging techniques like ultrasound, CT, magnetic resonance imaging (MRI), positron emission tomography (PET) are used for pancreatic cancer detection and staging. The most widely used technique is ultrasound, because it is inexpensive, safe and painless, it is recommended mostly for initial screening as it has fairly low sensitivity (67%) and specificity (40%). Ultrasound is not as convenient as CT, especially with multidetectors, intravenous contrast, curved planner reformations, and angiography. For example, helical computed tomography may detect pancreatic cancer masses larger than 2 cm with a high sensitivity of 78–100%. Despite some disadvantages of CT such as radiation expo-

sure and potential for contrast-induced nephropathy, it is a primary imaging method for pancreatic cancer diagnosis and staging.[LL14]

It is worth mentioning that MRI is superior to CT in the detection of small tumors and some other hard to notice changes in pancreas. On the other hand, MR image acquisition takes considerably longer time and it is more difficult to obtain uniform image quality.[SA10] Moreover, PET-CT with fluorine-18 fluorodeoxyglucose (18F-FDG) is a combination of PET and high-end multi-detector-row CT (MDCT), and is being widely used for diagnosis, staging, and monitoring of pancreatic cancer after treatment. 18F-FDG PET/CT can detect the metabolic activity in pancreatic cancer and evaluate tumor response to radiotherapy.[LL14]

Determining the stage of pancreatic cancer and assessing its resectability is usually done by using pancreatic protocol CT or CT angiography. Before performing these types of CT, iodinated nonionic contrast is injected intravenously to enhance image quality and visualize soft tissue. Arterial phase corresponds to first 30 seconds after injecting contrast, and provides opacification of the celiac axis, superior mesenteric artery, and peripancreatic arteries. Pancreatic phase images (35 to 40 seconds after contrast injection) show biggest enhancement and provide the best tumor to pancreas contrast.[LL14] Portal-venous phase (around a minute after contrast injection) provides better view of the superior mesenteric, splenic and portal veins, the pancreas itself and liver metastases if they are present. Pancreatic protocol CT is usually called the best CT imaging method for determining pancreatic cancer tumor resectability, but there is no sufficient evidence to support this.[TJA11] Phases and their times after contrast injection are listed in Table 1 table..

***1 table.*** *Pancreatic protocol CT phases*

| Phase | Seconds after contrast injection |
|---|---|
| Arterial | 20 |
| Pancreatic | 35-40 |
| Portal venous | 60-70 |

Pancreatic adenocarcinomas most often appear on CT as hypoattenuating masses (darker than surrounding tissues), but approximately 10% of pancreatic adenocarcinomas are isoattenuating (not directly visible) relative to the background pancreatic parenchyma, especially in small tumors 2 cm or less, thus making diagnosis difficult. In these situations, indirect signs, such as upstream pancreatic duct dilation can help diagnosis. The pancreas distal to the tumor usually appears atrophic. Growing tumor typically infiltrates the peripancreatic structures and may result in encasement of adjacent vasculature and in some cases adjacent organs. Pancreatic cancer tumors may have cysts or necrotic, and in rare cases they contain calcium.[LL14]

Figure 2 figure. shows an example of pancreatic cancer in CT. Lee and Lee (2014) provide CT image of a 73-year-old male patient with proven pancreatic head cancer. On the image 3 cm low attenuating mass is noted at the pancreatic head on the CT scan.


***2 figure.*** *Pancreatic cancer in CT*

Multidetector computed tomography (MDCT) is a type of CT that utilizes an array of detector rows to acquire image data simultaneously during a single rotation. Traditional CT scanners use single detector row and acquires smaller data volume with each rotation. MDCT is now routinely used for diagnosis and assessment of pancreatic cancer. It can display vascular anatomical variations and helps to better understand the features of the tumor. It also shows changes in tumor morphology and density, which helps to determine the stage of pancreatic cancer and its resectability. Also the degree of invasion to surrounding tissues and vessels can be assessed, with vascular involment being the most important factor for predicting resectability.[LL14] AS MDCT technology allows to decrease the influence of respiratory movement on image quality, it helps to find smaller tumors.[ZL20]

## 1.4 Medical Image Formats

DICOM (Digital Imaging and Communications in Medicine) image format is the international standard for medical images, that allows images to be exchanged with quality necessary for clinical use. DICOM is implemented in almost every imaging device (X-ray, CT, MRI, ultrasound, etc.), and is widely deployed healthcare messaging standards - there are billions of DICOM images in use for clinical care. The majority medical imaging datasets, both private and public, are in DICOM format. DICOM was first published in 1993, and since then it revolutionized the practice of medical imaging, allowing to fully digitalize and replace X-ray film. DICOM is approved by the International Organization for Standardization as the ISO 12052 standard[8].

DICOM specifies a nonproprietary biomedical images data interchange protocol, image format and file structure. The interface does not define just a hardware specification or a “plug and socket”, but the format and flow of the electronic messages of the images and related information between machines. If designed, configured and used properly to the DICOM specifications, equipment with a DICOM interface will communicate reliably with other DICOM equipment. Imaging system users have the freedom to select equipment based on functionality and quality rather than proprietary considerations because of DICOM interfaces available on nearly every model.[BHP+97]

Though DICOM is hugely successful, very flexible and comprehensive, it requires considerable effort and expense to implement correctly, and research environments usually lack the resources for ensuring this, which interferes with the advances in analysis methods and data. For this reason researchers usually use simpler image formats that allow to work easier and faster and only retain small relevant set of the images’ metadata, considering DICOM standard complicated. DICOM headers are often extended or customized by scanner manufacturers, which leads to duplication of data

[8] https://www.dicomstandard.org/about-home

and various incompatibilities in software that was only designed to work with specific format of DICOM.[LMA+16]

In contrast to DICOM, NIfTI (Neuroimaging Informatics Technology Initiative) format is very simple and minimalistic. CT scanners usually provide data in the DICOM format, but a lot of image processing tools used by scientists require images to be in the NIfTI format. These two formats are each suited for their specific purpose. DICOM is verbose and extensive, while NIfTI is very simple, and because of this, researchers often convert the images from DICOM to NIfTI, admitting that these two formats have their own purpose. Tools exists that allow images to be converted from DICOM to NIfTI, though there are multiple challenges to this process, and is sometimes requires customization.[LMA+16] In Medical Segmentation Decathlon 2018 dataset, data was converted from DICOM to NIfTI to ensure ease of access by enforcing consistency and interoperability, and avoid the need of proprietary software to view the images. Multiple scanners from multiple institutions were used to acquire the data, and the images were rotated to a standard anatomical coordinate system.[SAB+19] It should also be noted that there is little benefit in converting from NIfTI to DICOM, as in this case the file no longer has the granular and detailed information headers to take advantage of.[MPS+20]

NIfTI is not only used in neuroimaging, but also in other imaging fields. It is meant to work with the user and developer communities to address the need for compatibility and interoperability across tools and data. The initial focus of NIfTI was on tools that are used in fMRI, though it is now used for CT too.[9] Multiple software platforms can read and manipulate NIfTI images, including 3D Slicer[10], ITK Snap[11], and MATLAB[12]. There is also Python library, NiBabel, which has an API that gives full or selective access to header information (metadata), and image data is made available via NumPy arrays[13].

[9] https://nifti.nimh.nih.gov/background
[10] https://www.slicer. org/
[11] http://www.itksnap.org/
[12] https://www. mathworks.com/help/images/ref/niftiread.html
[13] https://nipy.org/nibabel/index.html

# 2 Keypoint Detection Algorithms

Some cancer classification and segmentation research uses machine learning and deep learning algorithms combined with classic computer vision algorithms. Extraction of important information from medical images is often done in several steps, which increases the accuracy of the methods. There is a compound of research done using various computer vision methods, combined with deep learning, classification methods like Naive Bayes or J48, logistic regression, etc. A subset of computer vision methods is designed to extract keypoints from the images. Keypoints are special points in images which are distinguished by their unique surroundings. Edges are another type of image features, which are found by edge detection algorithms, like Canny, Sobel and others.

There is a number of algorithms which are able to find keypoints in the analyzed images. The most known ones are Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), and Oriented FAST and Rotated BRIEF (ORB), with Features from Accelerated Segment Test (FAST) and Binary Robust Independent Elementary Features (BRIEF) also used independently. All of them are implemented in OpenCV library [14]. These algorithms and some others are described in the sections below. A research that applies these algorithms to medical images is also analyzed, but none of this research has been conducted on pancreas or pancreatic cancer.

Keypoint (or feature) detection finds important points in an image, like corners, edges, blobs, or regions, which can later be used for matching, tracking, or object recognition. To detect features, the algorithms scan the image and look for points that are stable under transformations like rotation or scaling. Some algorithms look for corners (e. g. Harris corner detection, FAST), others for blobs (e. g. SIFT using Difference of Gaussian, also KAZE and AKAZE), and some for edges (mostly for boundary detection, e. g. Canny edge detection). Sometimes keypoints also are assigned orientation. After this, a descriptor (vector) is computed around each keypoint, that describes the local image patch. The descriptors then can be compared, and this also allows to do matching between different images (for example, to stitch them of do object tracking in video).

In the following chapters the algorithms used in this Thesis are described in more detail.

## 2.1 Scale Invariant Feature Transform (SIFT)

Lowe [Low04] presented a method, Scale Invariant Feature Transform (SIFT), for extracting distinctive features from images, that are invariant to image scale and rotation, and can be used for robust matching. These features are very distinctive and each of them can be accurately matched against a database of features from a big count of images. Also they can be used for object recognition. Cascade filtering minimizes computational costs of the algorithm, as expensive operations are only used for specific features that pass a criteria. The steps of SIFT algorithm are as follows:

1. **Scale-space extrema detection.** At first algorithm creates scale space by progressively blurring the image with Gaussian filter multiple times. Then it searches for candidate features in subsequent layers by using a difference-of-Gaussian (DoG) function, which is approximation of

[14] https://docs.opencv.org/3.4/db/d27/tutorial_py_table_of_contents_feature2d.html

Laplacian-of-Gaussian. Keypoints are detected by locating local extrema in the DoG volume, comparing each pixel to its neighbors in 3x3 regions across scale space.

2. **Keypoint localization.** Each of candidate features are fit to a Taylor series expansion of scale space model, that determines their location and scale, and filtered by measures of stability so only high-contrast keypoints are left.

3. **Orientation assignment.** Each keypoint is assigned an orientation (or even several orientations) by calculating gradient directions and all subsequent operations (descriptor generation and matching) are performed having this orientation in mind, which provides invariance.

4. **Keypoint descriptor.** The area around keypoints is then inspected, and local image gradients are measured, which are later transformed into a vector.

SIFT is extensively used in medical image analysis, often in combination of some sort machine learning classification algorithms. Altaei and Kamil [AK20] proposed brain tumor classification system based on using SIFT for extracting features from MRI images, then used the extracted features to classify images as malignant or benign. The classifiers used were Naïve Bayes and J48, with results evaluated using cross-validation. Naïve Bayes had 98.9% accuracy for detecting brain tumors, and J48 had an accuracy of 100%, hence SIFT extracted features were deemed useful.

Piratla et al. [PJA20] proposed a method that uses mathematical morphology to improve contrast in MRI images. Wavelet transform with SIFT feature extraction and Bag Of Words (BOW) preprocess the images, which then are classified using Multi-Layer Perceptron. The method is compared to traditional SIFT technique of feature extraction, with classification accuracy of 94.82%.

Lal et al. [HSA+19] extracted texture, morphological, entropy based, Scale Invariant Feature Transform (SIFT), and Elliptic Fourier Descriptors (EFDs) features from brain tumor MRI images and used them to detect cancer using either Support Vector Machine (SVM) with polynomial, Radial Base Function (RBF), Gaussian kernels, Decision Tree (DT), or Naïve Bayes. The results were evaluated using Jack-knife 10-fold Cross-Validation (CV). They achieved 100% of accuracy using Naïve Bayes classifiers based on entropy, morphological, SIFT and texture features. Next was Decision Tree classifier with texture features that achieved accuracy of 97.81%, and SVM polynomial kernel with texture features with accuracy of 94.63%.

Kashif et al. tried to compare various keypoint detection and description algorithms (SIFT, SURF, BRIEF, BRISK, and FREAK) for bone age assessment. They applied sparse and dense feature points for each algorithm and created descriptors from keypoints within the epiphyseal regions of interest (eROI). The descriptor data was then used to perform classification using a support vector machine. SIFT with dense keypoints gave best results and outperforms all other algorithms, with accuracy of 98.36% within the range of 2 years.[KDH+16]

## 2.2 Speeded Up Robust Feature (SURF)

Bay et al. [BTV06] introduced an algorithm that speeds up SIFT and called it Speeded Up Robust Features (SURF). Bay et al. considered SIFT the most useful keypoint detection algorithm at the time,

but admitted that it has high computational requirements and has to be optimized in this regard, without sacrificing performance. SURF is very similar to SIFT but uses Box Filter instead of Laplacian-of-Gaussian for approximation with difference-of-Gaussian. It is based on the Hessian matrix and relies on integral images to speed up computation. It uses Haar-wavelet responses in both directions to describe the selected features. The dimension size of the features is reduced to 64 instead of 128. An additional indexing step based on sign of the Laplacian increases the robustness of the algorithm.

Pranata et al. [PWW+19] compared two CNN architectures - Residual network (ResNet) and a Visual Geometry Group (VGG) - for the classification performance of CT scans of bone fractures into fracture and non-fracture categories based on various views. The algorithm combined neural networks with fracture area matching using SURF method, Canny edge detection, and contour tracing. ResNet had a similar accuracy (98%) to the VGG network for classification but achieved better performance. ResNet outputs were used as the input for detecting the location and type of bone fracture using SURF.

Nguyen and Truong [NT19] used SURF as a descriptor for chromoendoscopy images. The features extracted were then used by Support Vector Machine to classify the images into either with gastrointestinal tract cancer or without. The accuracy of the proposed method reached 85%, with AUC of 0.92. The dataset used in this study was obtained with Olympus CV-180 endoscope at the Portuguese Institute of Oncology Hospital in Porto, Portugal.

Amalina et al. [ARS19] tried to build a system that searches histopathology image of colon cancer for the centroid position of nuclei, and classify it into inflammatory or non-inflammatory. SURF was used in this system for feature extraction, and the features were used by either k-Nearest Neighbor, Random Forest, or State Vector Machine. Nuclei detection F1 score was 0.722 with Determinant of Hessian thresholding with 50 as a parameter, which can be considered as average result, and nuclei classification best result was with Random Forest F1 score of 0.527, which is too low.

Ula et al. [UMS22] used SURF feature extraction in combination with multilayer perceptron to classify lung X-ray images into having pneumonia or not. They tried different momentum and learning rate values, and achieved best results with momentum of 0.1 and learning rate of 0.05, with accuracy of 0.827. The authors conclude that the research should be redone with different Multilayer Perceptron architectures, increasing or decreasing the number of hidden layers, outputs, etc., as better computational efficiency and accuracy could possibly be achieved.

## 2.3 Features from Accelerated Segment Test (FAST)

Rosten and Drummond [RD06] stated that feature detection methods that existed at the time, like Harris corner detector, SIFT and even SURF, were too computationally intensive for use in real-time applications, hence they proposed a faster algorithm, Features from Accelerated Segment Test (FAST). They also acknowledged that a high-speed detector is not very useful if the features it produces are unsuitable for processing.

FAST works by inspecting a circle of 16 pixels around the corner candidate. It classifies the candidate as a corner if there is a set of n adjacent pixels in the circle which are either all brighter than the intensity of the inspected pixel $I_p$ plus a threshold t, or all darker than $I_p$–$t$. $n = 12$ because

it is fast and excludes a lot of non-corners as it checks only four pixels - top, botto, left, and right. If candidate is a corner then at least three of these pixels must meet the criteria described earlier. Same procedure is repeated with other candidate pixels. This method has several weaknesses: FAST does not generalize well for n smaller than 12, there are implicit assumptions about the distribution of features when choosing the fast test pixels, the knowledge from the first 4 tests is discarded, and often adjacent features are detected. These problems are solved by either applying machine learning or non-birary suppression. In machine learning part, a decision tree is created that classifies the corners in the training dataset. It is then converted into C-code, creating a queue of nested if-else statements. FAST does not compute a corner response function, hence non-maximal suppression can not be applied directly to the features it finds. To solve this, a score function is computed for each feature, then corners which have a nearby corner with a higher score are removed by usin non-maximal suppression.[RD06]

FAST is being used in medical image analysis combined with machine learning methods. Sahi et al. [SMG+22] attempted to identify COVID-19 from a CT scan of the lungs. They first extracted features using FAST algorithm, and then used decision tree to classify images as COVID-19 or non-COVID-19. FAST was chosen because of its performance and speed. The algorithm was found to achieve high accuracy (0.98) using very few training examples.

Kusi-Duah et al. [KAA22] compared and evaluated several medical images texture feature extraction techniques, with FAST and FAST with BRIEF among them. FAST has yielded best F1 scores out of all used algorithms, though precision and accuracy was better for Local Binary Pattern (LBP) technique.

Dureja nad Pahwa [DP23] coupled CNN (VGG16) with FAST descriptor to extract an efficient low level and high-level features from the pre-data and match the features on the basis of similarity, measured by Minkowski distance, to make medical images retrieval from datasets as simple as possible. They used images from brain tumor, lung and liver disease datasets, available at Kaggle[15]. They achieved F1 score of 0.97 for the first dataset, 0.92 for the second dataset, and 0.76 for the third dataset.

## 2.4 Oriented FAST and Rotated BRIEF (ORB)

Rublee et al. [RRK+11] proposed even faster rotation invariant and noise resistant binary descriptor based on FAST and BRIEF. FAST algorithm was described earlier in this work, and BRIEF is a feature descriptor - it doesn't find the features in the image, but describes the features that are already found by some other algorithm. Rublee et al. conducted several experiments to demonstrate that ORB is twice as fast as SIFT with similar performance, and tested the algorithm on real-world applications. The method involves multiple steps:

1. An orientation component is added to FAST-9, and is applied to the image. The features found are then filtered by Harris corner measure.

[15] https://www.kaggle.com/

2. Intensity centroid measure is the applied to the features, and resulting vector is then used to impute an orientation.

3. BRIEF is steered according to the orientation of keypoints.

4. Steered BRIEF causes loss of variance, so a learning method was developed to choose a subset of binary tests with reduced correlation. The search is conducted among possible binary tests to find the ones with high variance and uncorrelated.

Andrianova and Demidova [AD21] developed four-stage approach to the matching of medical images (abdominal CT scans) based on the use of SIFT and ORB algorithms. First, SIFT and ORB algorithms are used to find keypoints and their descriptors for a pair of images. Then the descriptors are clustered to determine an optimal number of clusters or even find target clusters, and the centroids and medoids of the clusters are found. Then the matches are found based on Lowe's ratio test. Homography method is realized for the remaining keypoints and some of them are discarded as noise.

Shabbir et al. [MAA21] attempted to identify glaucoma in medical images (fundus pictures) using ORB. The algorithm was trained on 160 images of retinas, half of which were normal and half were glaucoma-affected. Green channel is extracted from colored fundus image, histogram equalization intensified the features of the images, ORB extracts the features which then are used by Naive Bayes classifier to differentiate between glaucoma and non-glaucoma images. The authors achieved precision and accuracy of 90%.

Atunnisa and Sumiharto [AS22] used ORB feature extraction for classification with CNN to detect pneumonia in lung X-ray images. They compared three methods - pure CNN, CNN with ORB feature detection and CNN without convolution layer with ORB feature detection. Two different datasets - two-class and three-class - from Kaggle were used for training and testing the models. CNN only result had accuracy of 96.3% for two classes and 82.1% for three classes dataset. CNN with ORB achieved accuracy of 60.17% for two classes and 53.1% for three classes. CNN without convolution layer with ORB had accuracy of 64.93% and 55%, respectively. Although accuracy results were not satisfying, the last method compared to pure CNN was 23 times faster.

## 2.5 Binary Robust Invariant Scalable Keypoints (BRISK)

BRISK is a computer vision algorithm that performs keypoint detection, description, and matching. Its primary benefit against other similar algorithms is computational efficiency. BRISK keypoint detection works by processing the scale-space pyramid layers, which consist of octaves and intra-octaves, created by halfsampling the original image (except for the first intra-octave, which is obtained by downsampling the original image by a factor of 1.5 instead of 2). BRISK applies a 16 pixel circle mask to each octave and intra-octave, and requires at least 9 consecutive pixels in it to be brighter or darker than the center pixel by a measure of some specified threshold. This technique is called FAST 9-16, and identifies regions of interest.[LCS11]

In the next step, non-maxima suppression in scale-space is applied to these regions. A point of the region is considered a keypoint only if its FAST score is higher than the scores of its 8 neighboring

points in the same layer. It must also have a higher score than the corresponding points in the octave layers above and below. To check this, a small square region o 2x2 pixels is examined around the point. Since different layers may have different resolutions, interpolation is used to make comparisons accurate. The detection of maxima across the scale axis at the first octave is a special case, as FAST 5-8 is applied instead of FAST 9-16, and scores for the next layer are not required to be lower.

After finding point of interest, an attempt is made to pinpoint exactly where the most important points are in an image — not just in terms of position, but also in scale. Instead of just picking the best point roughly, keypoints are refined to a more precise, sub-pixel level. Saliency is treated as something that changes smoothly not just across the image, but also across different scales. For each keypoint a 3x3 patch of scores (from FAST corner detection) is created in each of three octaves: the one where the keypoint was found, one layer above, and one below. Then in each layer, those scores are fit a 2D quadratic function to get a more accurate idea of where the maximum score is inside each patch. After refining within each layer, the three results are used to fit a 1D parabola along the scale axis. Finally, the the keypoint's position is adjusted by interpolating between the patches in the layers next to it.

Ouddai et al. applied BRISK, ORB and BRISK/DAISY to RGB histological images of breast cancer to recognise the tumor type, benign or malignant. Feature extraction algorithms are combined with BoF by kmeans and support vector machines. Color-BRISK/BoF/SVM solution had the best accuracy value (72.5%) while Color-ORB/BoF/SVM was the fastest.[OHG23]

BRISK was also used for bone age assessment in the already mentioned Kashif et al. study. In this study, SIFT outperformed all other algorithms, also BRISK.[KDH+16]

## 2.6 KAZE

KAZE is named in tribute to Iijima, the father of scale space analysis, and means wind in Japanese. Wind is a flow of air, ruled by nonlinear processes, and KAZE uses nonlinear diffusion processes in the image domain. Algorithms like SIFT and SURF use Gaussian blur to analyze images at multiple scales, but it tends to blur edges too much. KAZE uses nonlinear diffusion to smooth the image, which preserves edges and other important features, while smoothing flat areas.[ABD12]

In the first algorithm part, nonlinear scale space is built from original image up to a maximum evolution time using Additive Operator Splitting (AOS) techniques and variable conductance diffusion. Original image is convolved with a Gaussian kernel of standard deviation to reduce high-frequency noise. The scale space is discretized logarithmically into octaves and sublevels. Since nonlinear diffusion is defined in time, each scale is converted to time. The nonlinear diffusion partial differential equation is used and solved numerically.

Then, 2D features of interest are detected that exhibit a maxima of the scale-normalized determinant of the Hessian response through the 3×3×3 scale space. Then their position is refined using sub-pixel interpolation. Finally, the main orientation of the keypoint is computed by sampling first-order derivatives in a circular region of radius 6σ and weighting the vector using a Gaussian centered at the keypoint. A scale and rotation invariant descriptor is obtained considering first order image derivatives.

Li et al. attempted to categorise a collection of video images of echocardiograms, using KAZE coupled with the bag of words (BOW), sparse coding, and Fisher vector (FV). Compared to sparse SIFT which gives 72% overall accuracy on the classification of 8 viewpoints, KAZE with either BOW, sparse coding or FV improves the performance significantly, up to 81.09%, 78.85% and 80.8% respectively. When attempting to distinguish only 3 primary view locations, 97.44% accuracy can be achieved, compared to 90% accuracy with SIFT.[LQL+15]

Mohan et al. attempted to detect exudate (an indicator of diabetic retinopathy) in retinal images. They state that SIFT and SURF methods cannot differentiate features in the low contrast images, hence are not particularly suitable for the task. Authors used modified KAZE features to extract features for use with extreme learning machine autoencoders (ELMAE) for localization of the exudate. First the images are pre-processed, then optic disk is localized, dimensionality reduction performed using ELMAE, and finally exudate is localized. The sensitivity, specificity, and accuracy are 96.5%, 96.4%, and 97%, respectively, and processing time per image takes 3.19 seconds.[MMG+22]

Morillo et al. evaluated the performance of a pattern recognition system based on KAZE features in combination with Bag-of-Features (BOF) to discriminate between benign and malignant breast tumours in histopathological images. First, they extracted KAZE features from every image in the training set, then mapped the keypoints into a histogram vector using K-means clustering, which represents the input for a binary Support Vector Machine (SVM) classifier. When testing, the KAZE keypoints were extracted and mapped to a histogram vector by a cluster model. The vector was finally fed into the binary SVM classifier.[SGG+18]

## 2.7 Accelerated KAZE (AKAZE)

KAZE algorithm described above is slow because of the need to compute the nonlinear scale space, and Accelerated KAZE (AKAZE) is a modified version of KAZE, which is faster and more efficient. AKAZE uses three computational optimizations: replaces AOS with Fast Explicit Diffusion (FED) schemes, computes the determinant of the Hessian for each of the filtered images in the nonlinear scale space during feature detection, and proposes a Modified-Local Difference Binary (M-LDB) during feature description that exploits gradient and intensity information from the nonlinear scale space. As a result, AKAZE is several orders of magnitude faster than KAZE, while having the same or in some cases even better performance.[Fer13]

Shamla et al. suggested two state-of-the-art handcrafted feature extraction techniques, ORB and AKAZE in combination with Bag of Visual Word (BOVW) to classify echocardiogram images. Speckle Reduction Anisotropic Diffusion (SRAD) was applied to these images before feature extraction. Support Vector Machine (SVM), decision tree, and random forest algorithms in combination with ORB achieved 96.5%, 76%, and 97.7% accuracy respectively. AKAZE with SVM, decision tree, and random forest achieved accuracy rate of 97.7%, 90%, and 99%.[SRK+23]

## 2.8 Adaptive and Generic Corner Detection Based on the Accelerated Segment Test (AGAST)

Mair et al. improved accelerated segment test (AST, which is also a basis for FAST) by finding the optimal decision tree and extending its configuration space. AGAST provides higher performance for arbitrary environments, compared to FAST. Unlike FAST, AGAST algorithm is not trained for a specific scene, but it adapts to each different environment during image processing.[MHB+10]

AGAST chooses one of the pixels to test and one question to question to answer at a time, and the response is used to decide the following pixel and question. This changes searching for a corner from traversing ternary tree as in FAST (which asks three questions at a time - is the pixel brighter, darker or equal to the center pixel) to traversing a binary tree. Additionally the configuration space is increased by the adding two more states: “not brighter” and “not darker”. The binary decision trees reduce the entropy more quickly than a ternary tree, because less informative questions are performed in later stages of the decision process.

The optimal decision tree is found by exploring the configuration space from the root, where none of the pixels are known, then nodes of the tree are created while recursively evaluating one of the questions at a specific pixel. Depth First Search is performed until a leaf is found, and a leaf is defined as the first node which fulfills or can not fulfill anymore the criteria of corner.

AGAST also builds two trees, one of which specializes in homogeneous and one in structured regions. At the end of each decision path (i. e. when the corner criteria if met or not met) a jump to the appropriate specialized tree is done, depending on the configuration of this first leaf. This switch requires no additional computational cost, because the evaluation of the leaf node is already done when the specialized tree was being created. Using this technique any learning becomes needless, which improves efficiency of the algorithm.[MHB+10]

There are no existing papers that use AGAST algorithm for medical image analysis in the context of cancer detection. The existing studies only apply it to image registration and matching, hence it is not described in detail here.

# 3 Research and Results Analysis

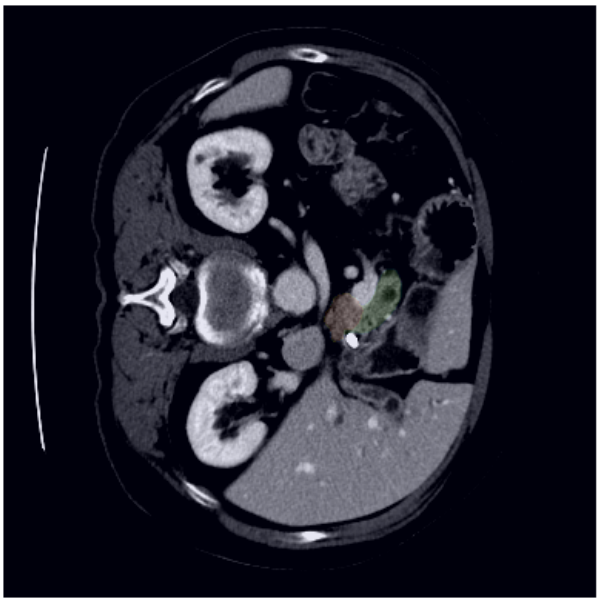
## 3.1 Detailed Research Plan

The detailed research plan is as follows:

1. Preprocess the dataset images.
2. Apply the following algorithms to the images in the dataset to find keypoints:
    - SIFT
    - SURF
    - FAST
    - ORB
    - BRISK
    - KAZE
    - AKAZE
    - AGAST
3. Perform parameter optimization for each of the algorithms to find parameters that create the most precise keypoints.
4. Compare the results of each algorithm with best performing parameters.

## 3.2 Image Processing

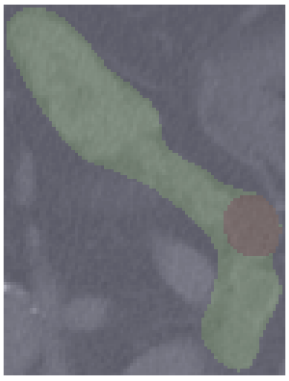
There are 420 NIfTI images in the Medical Segmentation Decathlon Pancreas dataset. Each image is 3-dimensional and consists of various number of slices (100 slices per image). The size of each slice is 512 x 512 pixels. Each image has a corresponding file with labels of pancreas and pancreatic tumor. Figure 3 figure. provides an example of a dataset image with its labels, that segment the pancreas and the tumor. It shows the CT slice of abdominal area of a patient, facing right. Near the center of it there are opaque areas marked in two colors, orange and green, and segment the area of pancreas (green) and pancreatic tumor (orange).



***3 figure.*** *Example of a dataset image with label overlay*

Nibabel library was used to open images of NIfTI format. The library function would return a 3-dimensional array of the size 512 x 512 x N, which then could be iterated to process and preview each slice separately (using matplotlib library). Retrieved slices are all grayscale. Pixel values for the images are floats in the range from -1 to 1, hence normalization was applied to convert the values to integers from 0 to 255 by using cv.normalize() function. Each image's label is stored in a separate file, also consisting of a 3-dimensional array of data. Labels are masks, where each pixel is assigned a value corresponding to either background (0), pancreas (1) or tumor (2). Besides the image data, nibabel object provides image metadata. There are some dimensions and offset metadata in the provided headers, which will not be used as it provides no benefit for the current tasks.

The images were further processed. The slices were examined and only the slices with a tumor in them were used. Random 198 slices were selected for the test. Then each slice label was inspected and 4 pixel coordinates were found, which correspond to the top, bottom, left and right of the pancreas or tumor in the image (anything that is not background). Using these coordinates the images were cropped to pancreas (the algorithms do not require that all images are of the same size). Figure 4 figure. provides examples of cropped images with their labels (pancreas and tumor).



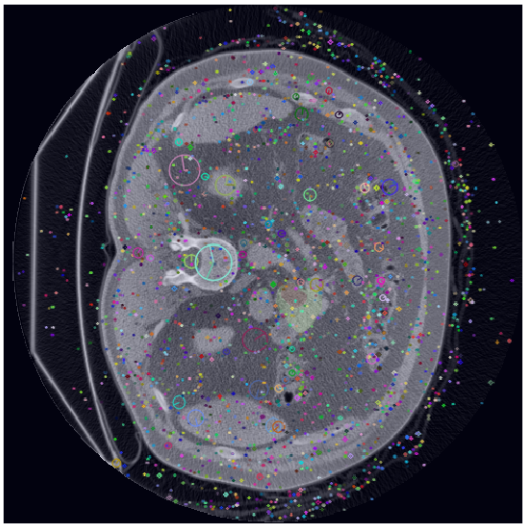
***4 figure.*** *Examples of cropped images with label overlay*

These cropped images were saved in separate files and loaded into memory before performing parameter tuning to save time during algorithm runs. Only several random slices out of each 3D NIfTI image is chosen, because keypoints in the image are usually stable despite transformations, hence subsequent slices have keypoints in the same regions, so if algorithm has not found keypoints inside tumor region in one of the slices, it is highly likely that it will also not find keypoints in a subsequent slice, and vice versa. The fraction of tumor pixels versus all pixels in the images is around 0.45, otherwise it is hard to evaluate the effectiveness of the algorithms due to high imbalance.

## 3.3 Evaluation Metrics

There are several metrics that evaluate keypoints: Mean Average Precision, Localisation Recall Precision, etc.[OCK+21] These metrics all assume that the ground truth is also a set of keypoints, but in our case there are no keypoints in the images of the training dataset but a labeled region which corresponds to either pancreas or tumor. One way to evaluate the task would be to draw keypoints in the training images, but which points should be chosen? It is not as obvious as in e. g. human posture recognition tasks, where keypoints correspond to human body joints, or looking for corners in an image. Choosing arbitrary points in the labeled zone and then comparing each keypoint to them would penalize the keypoints that are still in the labeled region but farther from the keypoint, and this is not a preferable outcome.

Another way would be to approach the problem as classification. Simplest way would be to count the images that have keypoints in the tumor region and calculate accuracy, but the problem with this approach is that different algorithms and even the same algorithms with different parameters create different counts of keypoints that can range from zero to hundreds of keypoints for smaller images, and to thousands of keypoints for bigger images (512 x 512). The more keypoints are in the image, the more likely there will be a keypoint in the tumor region, and the less keypoints there are, the less likely there will be a keypoint inside the tumor region (or any region). Figure 5 figure. provides an example of an image with a big number of keypoints. The metric should account for this and capture not only sensitivity, but also specificity of an algorithm.



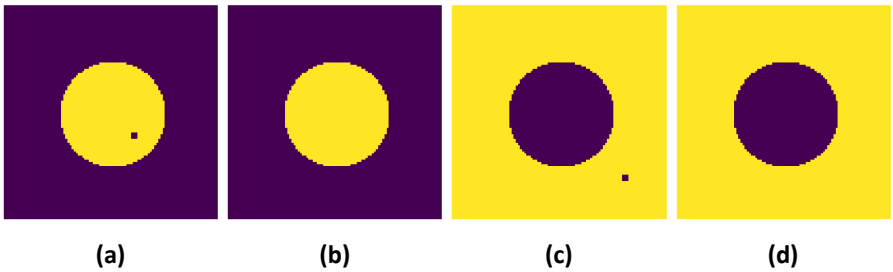
***5 figure.*** *Example of an image with 2576 keypoints*

It is possible to calculate the fraction of the keypoints inside the tumor region versus all keypoints. This metric is more meaningful, but still does not evaluate the results well because there is still an obvious relation between the total number of the keypoints and the metric. More sensitive algorithms that find a lot of keypoints usually have greater fraction of keypoints inside the tumor region, because less sensitive algorithms are less likely to find any keypoints in each image inside the tumor region and hence get zero score. It may imply that more sensitive algorithms are better, but it still does not provide any information on specificity and does not allow to discriminate between better performing algorithms and algorithms that just generate a lot of random keypoints. And also the goal of the study is not to find as many keypoints in the tumor region, but to find the keypoints inside the tumor region, which are specific to that region.

It is possible to count each keypoint in the tumor region as True Positive, and each keypoint outside of it as False Positive. Then the absence of a keypoint in the tumor region in an image could be counted as False Negative, and an absence of keypoints in the background as a True Negative. This can be treated as a highly imbalanced dataset and the metrics that deal with these types of datasets could be used, like balanced accuracy, Youden index, geometric mean of sensitivity and specificity, etc. Still, this method is not acceptable as it compares keypoints, which can be multiple in an image, with the absence of the keypoints, which can not be reasonably quantified.

If we introduce images without the tumor (only pancreas) and try to derive True Positives, True Negatives, False Positives and False Negatives, we have a problem that there is not one, but multiple keypoints in the image. The image without the tumor and without keypoint could be counted as True Negative. But one keypoint can be inside the tumor region, while another keypoint outside the tumor region. Should we treat this image as True Positive, as it found a keypoint inside tumor region, or as False Positive, because there are keypoints outside the tumor region too? The image without the tumor having keypoints and the image with the tumor but with keypoints only outside the tumor should have different meaning, but does it really say a different thing about the algorithm? This approach does not seem reasonable.

But there is one more way to approach the problem. OpenCV allows you to apply a mask to an image and look for keypoints inside that mask. Each image can be separated into two images by applying a mask to it - first by applying the mask to the tumor region and searching for the keypoints inside that mask. If keypoints are found, then it can be counted as True Positive, if keypoints were not found - as False Negative. Then the mask is applied to the region outside the tumor and the same algorithm looks for the keypoints there. If it finds any keypoints, the case is counted as False Positive, if not found - as True Negative. This approach captures not only algorithm sensitivity by evaluating if it succeeded to find any keypoints inside the tumor region, but also specificity by evaluating False Positives and True Negatives. It also does not favor algorithms that create a lot of keypoints and distinguishes them from useful algorithms. Figure 6 figure. provides a visual explanation of the metric evaluation. Yellow field is a mask in which the algorithm looks for the keypoints, while dark area is not considered. Note that there can also be multiple keypoints inside a mask, not only one.




***6 figure.*** *(a) True Positive (b) False Negative (c) False Positive (d) True Negative*

This method would have exact number of positive and negative examples and commonly used metrics like accuracy and F1 score can be used to evaluate the algorithms:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F1score = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

But there is also hidden imbalance here - the fraction of tumor label pixel versus all pixels is 0.1 on average if we take all dataset images, which means that though there is equal number of positive and negative examples because of application of masks to the images, the probability of a keypoint being in the tumor region is 0.1, and not 0.5, as we would expect in a classification problem with two classes. Therefore, we need to create such a dataset, when the probablity is closer to 0.5 by eliminating images with very small tumor region. This obviously makes the dataset different from real world data, and hence the results cannot be compared to results of other studies with different datasets.

For the imbalanced datasets metrics such as balanced accuracy and geometric mean of sensitivity and specificity are used:

$$sensitivity = \frac{TP}{TP + FN}$$

$$specificity = \frac{TN}{TN + FP}$$

$$balancedaccuracy = \frac{sensitivity + specificity}{2}$$

$$geometricmean = \sqrt{sensitivity \times specificity}$$

Balanced accuracy is an arithmetic mean between sensitivity and specificity, and can be used to measure performance of the algorithm in this case, but the geometric mean has a useful property that it is equal to 0 if either sensitivity or specificity is 0. For example, if after applying an algorithm to the dataset we have TP=198, TN=0, FP=198, FN=0, then sensitivity is 1, and specificity is 0. Accuracy is 0.5, F1 score is 0.667, balanced accuracy is also 0.5, but geometric mean is 0. It helps to discard these useless cases.

## 3.4 Results

The procedure of investigating algorithms is as follows: images from Medical Segmentation Decathlon dataset were one by one opened using nibabel library. Then each slice of the image was investigated whether it has a label corresponding to the tumor (integer 2 in label mask; 0 is background and 1 is pancreas). Only slices that have tumor in them are used for the task, as the mask will be applied to the images to create a set of images, meant to measure True Negatives. The images are cropped to pancreas, and images with very small tumor region are discarded.

After filtering the slices, each algorithm is applied to 1) each image with a mask filtering out the background and pancreas, to count True Positives and False Negatives 2) each image with a mask filtering out tumor region, to count True Negatives and False Positives. Then the resulting keypoints are analyzed. The main metric to compare is geometric mean of sensitivity and specificity, but other metrics like balanced accuracy, fraction of correctly identified images and average count of keypoints per image are also displayed. There is a tendency that the more keypoints are in the image, the bigger probability that some of these keypoints will be in the tumor region. Some of the algorithms' parameters influence the count of keypoints generated by lowering the threshold which tells whether the difference between pixels is meaningful. The lower the threshold, the more sensitive the algorithm becomes, and the more keypoints it detects. Hence various combinations of the parameters yield different counts of keypoints. The most desirable result is when there are as little keypoints as possible, but at the same time they are in the tumor region, and not outside of it. With some combinations of parameters, there are no keypoints in tumor region at all, but some parameter combinations can yield hundreds of keypoints.

To find the combination of parameters which yields the highest geometric mean, parameter tuning was performed for each algorithm using optuna[16] library. The library is "an open source hyperparameter optimization framework to automate hyperparameter search", which is efficient, easy to use and has many useful features like different samplers (including autosampler, that can automatically choose the most appropriate parameter sampling method), stores the results in a lightweight database, and has a visualization dashboard.[ASY+19] At the first stage each parameter was evaluated separately on a small number of images to see which parameter ranges yield meaningful results. Then parameter tuning was performed by running optuna study for each algorithm.

As parameter tuning can take quite a lot of time with a big count of images, some optimizations were added to the process. Image slices were preprocessed, saved, then opened and stored in memory before iterating through all of the parameter combinations. Preliminary experiments were done

[16] https://optuna.org/

with each algorithm parameters to find meaningful parameter ranges, which can also be considered an optimization that saves time. Also, instead of applying the algorithm twice and using masks, it can be applied once, but the keypoints inside the tumor region and outside of it can be counted separately. Another optimization was to run the algorithm on multiple threads. On a 12 core machine 10 threads were used, by dividing the data into 10 parts and running each iteration in parallel. Still, the size of the images impacted the speed the most. All these optimizations brought the times of running the parameter optimization for each algorithm from hours to minutes.

Algorithms were run on the system with the following configuration:

- MacBook Pro with MacOS 13.6.1, 2,6 GHz 6-Core Intel Core i7, 32 GB RAM
- python 3.9.18 with anaconda 22.9.0
- OpenCV v4.8.0

The code for running the algorithms is available. It contains several Python notebooks, each of which uses one of the algorithms to perform parameter tuning, and visualize the sample images with keypoints and labels. One of the notebooks contains the code to preprocess the dataset images. Note that the dataset has to be downloaded and extracted to a specified folder separately.

### 3.4.1 SIFT

SIFT was included in OpenCV non-free modules (opencv_contrib repository) until 2020, when the patent for SIFT has expired. In newer OpenCV versions, SIFT is included in the main repository, so building OpenCV from source is not required. SIFT object is created by calling cv.SIFT_create()[17], which also accepts multiple parameters:

- **nfeatures** - defines the number of best features to keep, corresponds to local contrast. Default is 0.
- **nOctaveLayers** - defines the number of layers in each octave, which is computed automatically based on the resolution. Default is 3.
- **contrastThreshold** - removes weak keypoints in low-contrast areas; decreases the number of keypoints found. When the filtering is applied, the parameter is divided by nOctaveLayers. Default is 0.04.
- **edgeThreshold** - the parameter is used to filter out edge features. The bigger the parameter value, the more keypoints are found. Defaults to 10.
- **sigma** - is the sigma of the Gaussian. It is applied to the image at the octave with number 0. It is advised to reduce it if the image is of low contrast. Default is 1.6.
- **enable_precise_upscale** - defines whether to enable precise upscaling. It prevents localization bias and is disabled by default (is False).

[17] https://docs.opencv.org/4.8.0/d7/d60/classcv_1_1SIFT.html

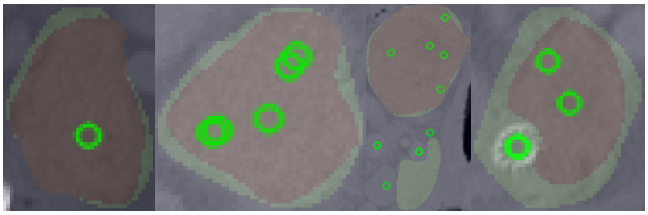
Keypoint detection is done by calling sift.detect(), which accepts two arguments - an image in which to detect the keypoints and a mask that limits image to specific region where to search for the keypoints. The second argument can be omitted due to optimizations mentioned earlier. Keypoints are visualized by calling cv.drawKeypoints().

After preliminary investigation of SIFT parameters the following ranges of parameters were used in the parameter tuning:

- **nfeatures** - this parameter seems to increase the number of keypoints found in the images, i. e. sensitivity of the algorithm. Range from 0 to 70 is used.
- **nOctaveLayers** - this parameter also seems to increase sensitivity of the algorithm, range from 1 to 200 is used here (0 gives validation error).
- **contrastThreshold** - range from 0 to 0.1 is used for this parameter, it decreases the number of found keypoints.
- **edgeThreshold** - the parameter increases the number of keypoints found in the image. At first the change is significant, and then it becomes small. Range from 0 to 70 is used.
- **sigma** - the parameter decreases the number of keypoints found in the image. With sigma=0 algorithm gives an error, in the range from 0.1 to 0.3 it finds a small number of keypoints, then at 0.4 it finds the biggest number of keypoints and since this value the count of keypoints gradually decreases. Range from 0.1 to 15.0 is used for parameter tuning.
- **enable_precise_upscale** - as it is either True of False, trying both.

Parameter tuning for this algorithm takes about 6 minutes using 198 images as the algorithm is quite slow compared to others. Best achieved geometric mean - 0.687, though balanced accuracy is higher (0.689). Best result has approximately 2.328 keypoints per image on average. Best result also has TP=126, TN=147, FP=51, FN=72, sensitivity=0.636, specificity=0.742, with keypoints per image ranging from 0 to 23, 2.328 on average. The fraction of keypoints in the tumor zone is 0.692. Algorithm specificity and sensitivity is pretty similar, and it creates a small number of keypoints in the images.

Table 2 table. provides a sample of 10 best results for SIFT. Some parameters vary (nfeatures, edgeThreshold, nOctaveLayers, sigma) and others (contrastThreshold, enable_precise_upscale) are the same or very similar for all best results.




***7 figure.*** *SIFT keypoint examples. a) and b) - with keypoints inside tumor region and no points outside of it; c) and d) with keypoints in both tumor region and outside of it*

| Rank | nfeat | sigma | contrThr | edgeThr | enable | nOctLayers | g-mean | b-acc | avg kp |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 38 | 8.272 | 0.003 | 12 | True | 8 | 0.687 | 0.689 | 2.328 |
| 2 | 16 | 9.635 | 0.001 | 8 | True | 76 | 0.686 | 0.687 | 6.035 |
| 3 | 51 | 9.167 | 0.001 | 14 | True | 31 | 0.684 | 0.687 | 2.515 |
| 4 | 50 | 5.527 | 0.002 | 25 | True | 2 | 0.683 | 0.684 | 3.596 |
| 5 | 53 | 8.123 | 0.001 | 13 | True | 30 | 0.679 | 0.679 | 3.384 |
| 6 | 48 | 8.110 | 0.003 | 15 | True | 33 | 0.679 | 0.679 | 3.308 |
| 7 | 49 | 8.326 | 0.001 | 15 | True | 33 | 0.679 | 0.679 | 3.338 |
| 8 | 21 | 7.474 | 0.001 | 19 | True | 19 | 0.679 | 0.679 | 3.197 |
| 9 | 50 | 9.825 | 0.002 | 22 | True | 58 | 0.679 | 0.679 | 5.798 |
| 10 | 65 | 9.914 | 0.002 | 17 | True | 10 | 0.678 | 0.689 | 1.722 |

***2 table.*** *Top 10 Best Trials for SIFT*

**(a)** **(b)** **(c)** **(d)**

***8 figure.*** *SIFT keypoint examples. a) and b) - with keypoints outside of tumor region and not inside of it; c) and d) no detected keypoints*

Figures 7 figure. and 8 figure. provide examples of SIFT keypoints in the images. There can be none or several keypoints in each image, though generally there are not a lot of keypoints. As keypoint count per image is not big, algorithms sensitivity and specificity are quite similar.

### 3.4.2 SURF

SURF[18] is a patented algorithm, and commercial use of it without license is prohibited[19]. Because of this, the algorithm is not included in the main OpenCV library, but only in opencv_contrib repository. OpenCV has to be built from source with opencv_contrib included and with CMake flag OPENCV_ENABLE_NONFREE set to ON. Then the algorithm is accessible in module xfeatures2d.

SURF object is created by calling cv.xfeatures2d.SURF_create() and accepts several different parameters:

- **hessianThreshold** - defines a threshold for the keypoint detector. Only keypoints with hessian value larger than the threshold are kept. The larger the value, the less keypoints are found by the algorithm. Default value is 100.
- **nOctaves** - defines the number of Gaussian pyramid octaves, with larger values giving larger features. Default value is 4.

[18] https://docs.opencv.org/4.8.0/d5/df7/classcv_1_1xfeatures2d_1_1SURF.html
[19] https://people.ee.ethz.ch/~surf/download.html

- **nOctaveLayers** - defines the number of images within each octave of a Gaussian pyramid. It is set to 2 by default.
- **extended** - defines whether to compute basic (64 elements) or extended (128 elements) descriptors. Defaults to 0 (basic).
- **upright** - defines whether to compute the orientation of a keypoint descriptor. upright=0 computes the orientation, and upright=1 does not, which makes the algorithm much faster. Default is 0.

The following SURF parameters were used for the parameter optimization:

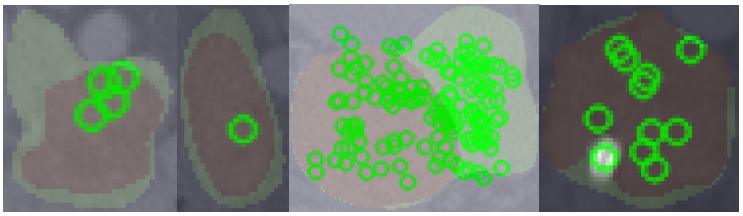
- **hessianThreshold** - this parameter decreases the number of keypoints found in an image. Range from 0 to 150 is used for the grid search.
- **nOctaves** - the parameter slightly increases the count of keypoints, range from 1 to 10 is used.
- **nOctaveLayers** - the parameter slightly increases the count of keypoints, range from 1 to 10 is used.
- **extended** - as only detectors are used, default value 0 is used (do not compute extended descriptors for faster algorithm runs).
- **upright** - this parameter defines the orientation of a descriptor, which will not be computed, hence using value 1 (do not compute the orientation).

Keypoint detection is done by calling surf.detect(), which, similarly to SIFT, accepts an image and a mask. The parameter optimization for this algorithm takes several minutes using 198 slices as there are very few parameters. Best achieved accuracy is 0.702, best geometric mean - 0.700. Best result also has TP=149, TN=129, FP=69, FN=49, sensitivity=0.753, specificity=0.652, with keypoints per image ranging from 0 to 266, 19.116 on average. The fraction of keypoints in the tumor zone is 0.709. As there a quite a lot of keypoints detected by the algorithm, sensitivity is higher than specificity. The algorithm does better finding True Positives than True Negatives.

| Rank | nOctaves | hessianThreshold | nOctaveLayers | g-mean | b-acc | avg kp |
|---|---|---|---|---|---|---|
| 1 | 8 | 0 | 6 | 0.700 | 0.702 | 19.116 |
| 2 | 9 | 0 | 8 | 0.700 | 0.702 | 19.490 |
| 3 | 9 | 0 | 7 | 0.700 | 0.702 | 19.364 |
| 4 | 1 | 0 | 10 | 0.700 | 0.702 | 18.510 |
| 5 | 1 | 0 | 8 | 0.700 | 0.702 | 18.429 |
| 6 | 1 | 0 | 9 | 0.700 | 0.702 | 18.470 |
| 7 | 1 | 0 | 5 | 0.700 | 0.702 | 17.682 |
| 8 | 5 | 0 | 8 | 0.700 | 0.702 | 19.490 |
| 9 | 10 | 0 | 10 | 0.700 | 0.702 | 19.571 |
| 10 | 1 | 0 | 6 | 0.700 | 0.702 | 18.061 |

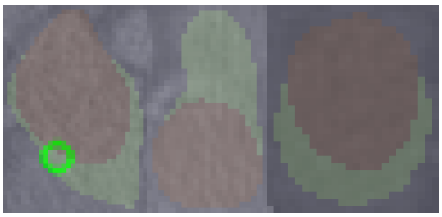
***3 table.*** *Top 10 Best Trials for SURF*

Table 3 table. provides a sample of best results in SURF parameter optimization. Note that these are not all best results as these repeat themselves in various nOctaves and nOctaveLayers ranges. Algorithm performs best with hessianThreshold value 0, and nOctaves and nOctaveLayers vary significantly.



(a) (b) (c) (d)

***9 figure.*** *SURF keypoint examples. a) and b) - with keypoints inside tumor region and no points outside of it; c) and d) with keypoints in both tumor region and outside of it*



(a) (b) (c)

***10 figure.*** *SURF keypoint examples. a) - with keypoints outside of tumor region and not inside of it; c) and d) no detected keypoints*

Figures 9 figure. and 10 figure. provide examples of SURF keypoints found in the images, visualized by calling cv.drawKeypoints(). There can be 0, 1 or multiple keypoints in each image, with keypoints detected in the regions of various area. Having more keypoints does not necessarily mean that a keypoint will also be detected in the tumor region, and if the region is big it also does not mean that a keypoint will be found there.

### 3.4.3 FAST

FAST[20] detector in OpenCV is created by calling cv.FastFeatureDetector_create() with the following parameters:

- **threshold** - defines the threshold value for the algorithm. Default value is 10.
- **nonmaxSuppression** - tells whether to use nonmaximal suppression. Defaults to true.
- **type** - defines which type of detector to use. The options are TYPE_5_8, TYPE_7_12 and TYPE_9_16. Default is TYPE_9_16.

[20] https://docs.opencv.org/4.8.0/df/d74/classcv_1_1FastFeatureDetector.html

The algorithm appeared to be considerably faster than the other algorithms, and able to run hundreds of times just in several seconds. This is due to his relative simplicity and computational efficiency, described in the chapter earlier. The following ranges of the parameters were taken for the parameter optimization:

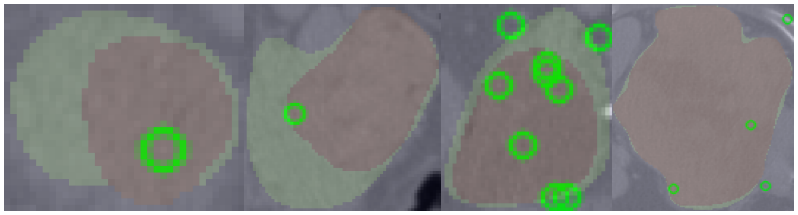
- **threshold** - range from 0 to 100 has been used for tuning. The parameter changes the count of keypoints in nonlinear way.
- **nonmaxSuppression** - as it can be either True or Talse, both values were used for the parameter optimization.
- **type** - also all possible options were used for the parameter optimization.

The optimization for this algorithm takes approximately half a minute to complete using 198 images as there are very few parameters and the algorithm itself is very fast. Best achieved geometric mean - 0.475, balanced accuracy up to 0.477. Best result also has TP=104, TN=85, FP=113, FN=94, sensitivity=0.525, specificity=0.429, with keypoints per image ranging from 0 to 56, 4.929 on average. The fraction of keypoints in the tumor zone is 0.542.

| Rank | type | nonmaxSuppression | threshold | g-mean | b-acc | avg kp |
|---|---|---|---|---|---|---|
| 1 | 0 | True | 3 | 0.475 | 0.477 | 4.929 |
| 2 | 0 | False | 3 | 0.475 | 0.477 | 5.025 |
| 3 | 0 | True | 2 | 0.437 | 0.515 | 15.530 |
| 4 | 0 | False | 2 | 0.437 | 0.515 | 16.146 |
| 5 | 0 | True | 4 | 0.410 | 0.452 | 1.742 |
| 6 | 0 | True | 5 | 0.333 | 0.452 | 0.758 |
| 7 | 0 | False | 5 | 0.333 | 0.452 | 0.793 |
| 8 | 1 | True | 5 | 0.326 | 0.470 | 26.960 |
| 9 | 2 | True | 6 | 0.292 | 0.424 | 18.616 |
| 10 | 2 | False | 9 | 0.285 | 0.295 | 18.843 |

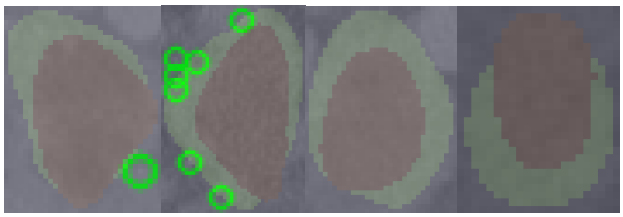
***4 table.*** *Top 10 Best Trials for FAST*

Table 4 table. provides a sample of best results in FAST parameter tuning. Algorithm performs best with small threshold values, and type of detector is TYPE_5_8. Overall results are quite poor with a fraction of images with keypoints in the tumor zone equal to 0.329. Balanced accuracy is sometimes much higher than geometric mean, with very high count of both False Positives and False Negatives.



**(a)** **(b)** **(c)** **(d)**

***11 figure.*** *FAST keypoint examples. a) and b) - with keypoints inside tumor region and no points outside of it; c) and d) with keypoints in both tumor region and outside of it*



(a) (b) (c) (d)

***12 figure.*** *FAST keypoint examples. a) and b) - with keypoints outside of tumor region and not inside of it; c) and d) no detected keypoints*

Figures 11 figure. and 12 figure. provide examples of FAST keypoints found in the images. The counts of keypoints in the images vary significantly. FAST seems to perform better on the images with greater area of tumor region, and perform really poorly on the images with small tumor region. It tends to capture various noise in the images.

### 3.4.4 ORB

As was already mentioned, ORB is an alternative to SIFT and SURF, which uses FAST keypoint detector and BRISK descriptor and adds various modifications, so it would be reasonable to expect similar performance to the previous algorithm. ORB is created by calling cv.ORB_create()[21] function that has the following parameters:

- **nfeatures** - similar to the other algorithms, this means the maximum number of features to keep, and defaults to 500.
- **scaleFactor** - defines pyramid decimaton ration, which should be between 1 and 2. Value close to 1 means more pyramid levels are needed to cover the image and the algorithm will become slower, but value of 2 means each higher level has 4 times less pixels, and this impacts feature matching score. Default value of 1.2 is a good balance between speed and quality.
- **nlevels** - defines number of pyramid levels and defaults to 8.
- **edgeThreshold** - defines the size of the border in the image where keypoints are not detected. It defaults to 31 and should approximately match patchSize parameter.
- **firstLevel** - the level in the pyramid with the source images, other pyramid layers are filled with upscaled image. Default is 0.
- **WTA_K** - this parameter is related to BRIEF descriptor part of the algorithm and defines the number of pixels that produce an element in descriptor. This is not relevant to the detector and hence will not be used in the parameter optimization. Default value is 2.
- **scoreType** - defines whether to use Harris or FAST score for keypoint ranking. Harris is more stable but slower, whereas FAST is faster but less stable. Defaults to Harris (cv.ORB_HARRIS_SCORE).

[21] https://docs.opencv.org/4.x/db/d95/classcv_1_1ORB.html

- **patchSize** - relates to BRIEF descriptor and defines the size of the patch. This will not be used in parameter optimization as only detector is tested here. Defaults to 31.
- **fastThreshold** - FAST threshold, defaults to 20.

The following ranges of the parameters were taken for the parameter optimization:

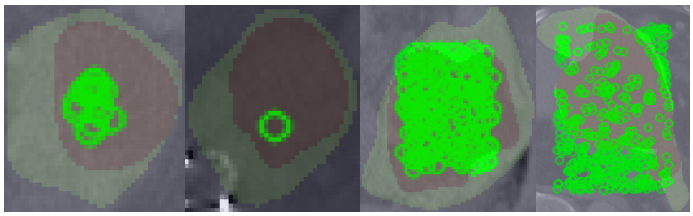
- **nfeatures** - range from 0 to 1000 is used, though this parameter does not seem to make significant impact on the keypoints found in the images.
- **scaleFactor** - this parameter very slightly impacts the keypoints found in the images. Range from 1.0 to 1.3 is used.
- **nlevels** - this parameter also barely affects the keypoints found by the algorithm. Values from 1 to 9 are suggested in tuning.
- **edgeThreshold** - range from 1 to 30 is used for this parameter, increasing the parameter seems to decrease the count of keypoints in an image.
- **firstLevel** - range from 0 to 6 is used for this parameter, it seems to decrease the number of keypoints found in the images. The parameter is not affecting the results significantly.
- **WTA_K** - this parameter does not impact the keypoints found in the image as it relates to the descriptor and not detector. Default value was used.
- **scoreType** - the score type also does not seem to impact the keypoints found, trying both FAST score and Harris score.
- **patchSize** - this parameter is related to descriptor instead of detector. Trying values from 2 to 60.
- **fastThreshold** - range from 1 to 40 is used for this parameter. Decreasing the value increases the number of keypoints found in the images, making the algorithm more sensitive.

Running parameter optimization for the algorithm takes around 1 minute. Though there are a lot of parameters, a lot of them do not significantly impact the results. Highest achieved geometric mean is 0.715, which is better than FAST. Balanced accuracy in the top 10 results reaches 0.717. Best result also has TP=132, TN=152, FP=46, FN=66, sensitivity=0.667, specificity=0.768, with keypoints per image ranging from 0 to 500, 58.984 on average, which is quite a lot compared to other algorithms. The fraction of keypoints in the tumor zone is 0.718.

Table 5 table. provides a sample of best results in ORB parameter tuning. The algorithm seems to perform well with scaleFactor ranging from 1.018 to 1.188, edgeThreshold from 16 to 23, and fastThreshold 1. Nlevels seem to have no significant impact on the result, as it can be any value in the selected range.

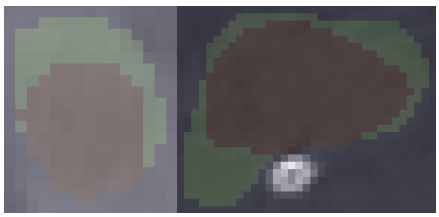
| Rank | fastThr | score | edgeThr | firstLvl | nlvls | scale | g-mean | b-acc | avg kp |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 18 | 1 | 2 | 1.055 | 0.715 | 0.717 | 58.985 |
| 2 | 1 | 0 | 16 | 0 | 4 | 1.083 | 0.712 | 0.712 | 69.101 |
| 3 | 1 | 0 | 16 | 0 | 1 | 1.003 | 0.712 | 0.712 | 43.470 |
| 4 | 1 | 0 | 22 | 2 | 4 | 1.188 | 0.712 | 0.712 | 76.268 |
| 5 | 1 | 1 | 23 | 3 | 7 | 1.109 | 0.710 | 0.712 | 95.601 |
| 6 | 1 | 0 | 16 | 0 | 2 | 1.089 | 0.710 | 0.710 | 59.944 |
| 7 | 1 | 0 | 17 | 1 | 1 | 1.018 | 0.710 | 0.712 | 29.232 |
| 8 | 1 | 0 | 19 | 1 | 3 | 1.141 | 0.709 | 0.710 | 63.929 |
| 9 | 1 | 0 | 18 | 0 | 5 | 1.137 | 0.709 | 0.715 | 49.944 |
| 10 | 1 | 0 | 18 | 0 | 1 | 1.071 | 0.709 | 0.715 | 36.424 |

***5 table.*** *Top 10 Best Trials for ORB*



**(a)** **(b)** **(c)** **(d)**

***13 figure.*** *ORB keypoint examples. a) and b) - with keypoints inside tumor region and no points outside of it; c) and d) with keypoints in both tumor region and outside of it*



**(a)** **(b)**

***14 figure.*** *ORB keypoint examples. a) and b) - no detected keypoints*

Figures 13 figure. and 14 figure. provide examples of ORB keypoints found in the images. The count of the keypoints in the images vary dramatically, but generally the algorithm creates a large number of keypoints. The pattern on some images looks quite spectacular as all keypoints form a square area.

### 3.4.5 BRISK

BRISK[22] object is created in OpenCV by calling cv.BRISK_create(), which accepts the following parameters:

- **thresh** - defines AGAST detection threshold score. Defaults to 30.

[22] https://docs.opencv.org/4.8.0/de/dbf/classcv_1_1BRISK.html

- **octaves** - defines detection octaves, with 0 meaning single scale will be used. Defaults to 3.
- **patternScale** - defines a scale of the pattern, which is used for sampling the neighborhood. Defaults to 1.

The following algorithm parameters were used for the parameter tuning:

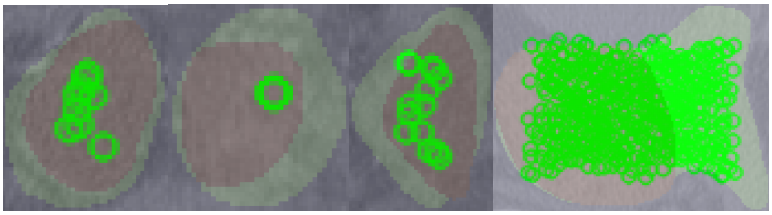
- **thresh** - range from 0 to 60. The parameter decreases the count of keypoints found in an image.
- **octaves** - range from 0 to 4, the parameter slightly decreases the count of keypoints found.
- **patternScale** - range from 0 to 1.6, the parameter decreases the count of keypoints found.

There are not many parameters, but the algorithm itself is quite slow, running the parameter tuning for the algorithm takes 9 minutes. Best achieved geometric mean is 0.726, highest balanced accuracy in top 10 results - 0.727. Best result also has TP=137, TN=151, FP=47, FN=61, sensitivity=0.692, specificity=0.763, with keypoints per image ranging from 0 to 1061, 60.449 on average. The fraction of keypoints in the tumor zone is 0.756.

| Rank | patternScale | octaves | thresh | g-mean | b-acc | avg kp |
|---|---|---|---|---|---|---|
| 1 | 1.097 | 2 | 0 | 0.726 | 0.727 | 60.449 |
| 2 | 1.097 | 2 | 0 | 0.726 | 0.727 | 60.449 |
| 3 | 1.083 | 3 | 0 | 0.726 | 0.727 | 60.722 |
| 4 | 1.095 | 3 | 0 | 0.726 | 0.727 | 60.515 |
| 5 | 1.092 | 3 | 0 | 0.726 | 0.727 | 60.636 |
| 6 | 1.092 | 3 | 0 | 0.726 | 0.727 | 60.636 |
| 7 | 1.092 | 3 | 0 | 0.726 | 0.727 | 60.636 |
| 8 | 1.092 | 3 | 0 | 0.726 | 0.727 | 60.636 |
| 9 | 1.091 | 3 | 0 | 0.726 | 0.727 | 60.636 |
| 10 | 1.095 | 3 | 0 | 0.726 | 0.727 | 60.515 |

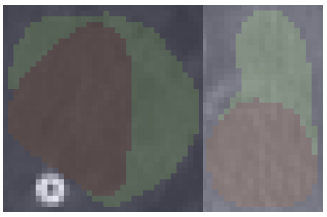
***6 table.*** *Top 10 Best Trials for BRISK*

Table 6 table. provides a sample of best results in BRISK parameter optimization. Algorithm performs best with threshold 0, octaves from 2 or 3 and patternScale from 1.083 to 1.097.



(a) (b) (c) (d)

***15 figure.*** *BRISK keypoint examples. a) and b) - with keypoints inside tumor region and no points outside of it; c) and d) with keypoints in both tumor region and outside of it*



(a) (b)

***16 figure.*** *BRISK keypoint examples. a) and b) - no detected keypoints*

Figures 15 figure. and 16 figure. provide examples of BRISK keypoints. The algorithm tends to create a lot of keypoints, with the counts of keypoints varying significantly. BRISK sometimes creates a lot of keypoints in a small region, forming a square, similar to ORB.

### 3.4.6 KAZE

OpenCV implements KAZE[23] algorithm in its module features2d. Algorithm object is created by calling cv.KAZE_create() with the following parameters:

- **extended** - similar to other algorithms, this parameter defines if extended (128-byte) descriptors should be used, defaults to false (use basic descriptors of size 64).
- **upright** - also is similar to other algorithms, defines if upright descriptors should be used. If set to False, the orientation of the keypoints is calculated, which makes the algorithm slower. Defaults to False (calculate the orientation of the keypoints).
- **threshold** - defines a threshold at which to accept a keypoint. Defaults to 0.001.
- **nOctaves** - defines maximum octave evolution of the image, defaults to 4.
- **nOctaveLayers** - defines number of sublevels per scale level, defaults to 4.
- **diffusivity** - defines diffusivity type, which can be one of DIFF_PM_G1, DIFF_PM_G2, DIFF_WEICKERT or DIFF_CHARBONNIER. Default is DIFF_PM_G2.

The following algorithm parameter ranges were used for the parameter optimization:

- **extended** - parameter is related to descriptors and is not needed for keypoint detection, so default will be used (True).
- **upright** - this parameter also is realted to descriptors, keypoint oriantation calculation should be turned off to save time (False).
- **threshold** - range from 0 to 0.002 will be used.
- **nOctaves** - range from 1 to 10 will be used.

[23] https://docs.opencv.org/4.8.0/d3/d61/classcv_1_1KAZE.html

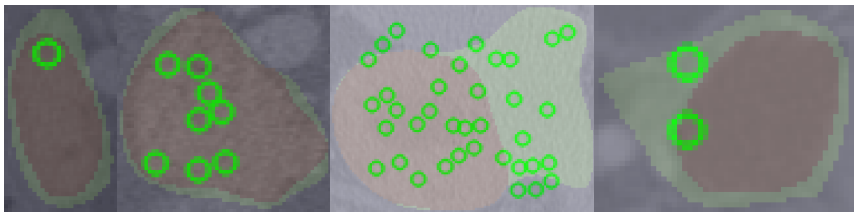
- **nOctaveLayers** - range from 1 to 10 will be used.
- **diffusivity** - all possible variations will be used.

Because there are few parameters, and the algorithm itself is quite slow, running the parameter tuning for the algorithm takes approximately 9 minutes. Best achieved geometric mean is 0.636, balanced accuracy - 0.636. Best result also has TP=128, TN=124, FP=74, FN=70, sensitivity=0.646, specificity=0.626, with keypoints per image ranging from 0 to 76, 6.359 on average. The fraction of keypoints in the tumor zone is 0.658.

| Rank | nOctaveLayers | nOctaves | diffusivity | threshold | g-mean | b-acc | avg kp |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 7 | 3 | 0.000 | 0.636 | 0.636 | 6.359 |
| 2 | 1 | 7 | 3 | 0.000 | 0.634 | 0.634 | 6.056 |
| 3 | 1 | 4 | 3 | 0.000 | 0.634 | 0.634 | 6.131 |
| 4 | 1 | 4 | 3 | 0.000 | 0.634 | 0.634 | 5.561 |
| 5 | 1 | 8 | 3 | 0.000 | 0.634 | 0.634 | 5.884 |
| 6 | 1 | 6 | 3 | 0.000 | 0.634 | 0.634 | 6.520 |
| 7 | 1 | 4 | 3 | 0.000 | 0.631 | 0.631 | 5.253 |
| 8 | 1 | 4 | 3 | 0.000 | 0.631 | 0.631 | 5.273 |
| 9 | 1 | 6 | 3 | 0.000 | 0.631 | 0.631 | 5.429 |
| 10 | 1 | 7 | 3 | 0.000 | 0.631 | 0.631 | 5.601 |

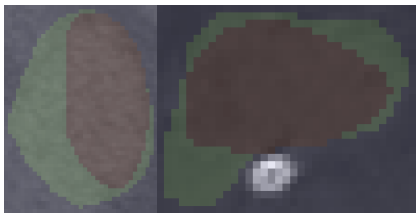
***7 table.*** *Top 10 Best Trials for KAZE*

Table 7 table. provides a sample of best results in KAZE parameter tuning. Algorithm performs best with very small threshold value, 1 octave layer, 4 to 8 octaves and diffusion value 3. Algorithm itself seems to be quite slow, but the count of the parameters is not high, so it finishes parameter tuning in 9 minutes.



**(a)** **(b)** **(c)** **(d)**

***17 figure.*** *KAZE keypoint examples. a) and b) - with keypoints inside tumor region and no points outside of it; c) and d) with keypoints in both tumor region and outside of it*



**(a)** **(b)**

***18 figure.*** *KAZE keypoint examples. a) and b) - with keypoints outside of tumor region and not inside of it; c) and d) no detected keypoints*

Figures 17 figure. and 18 figure. provide examples of KAZE keypoints found in the images. The count of keypoints in each image varies significantly. The fraction of correctly guessed images is slightly higher than chance, and similar to other algorithms.

### 3.4.7 AKAZE

OpenCV also implements Accelerated KAZE or AKAZE[24] algorithm in its module features2d. Algorithm object is created by calling cv.AKAZE_create() with the parameters, which are similar to KAZE, with some extra parameters related to descriptors:

- **descriptor_type** - defines the type of the descriptor, which can be one of DESCRIPTOR_KAZE, DESCRIPTOR_KAZE_UPRIGHT, DESCRIPTOR_MLDB or DESCRIPTOR_MLDB_UPRIGHT; defaults to DESCRIPTOR_MLDB.
- **descriptor_size** - defines the size of the descriptor, default is 0.
- **descriptor_channels** - defines the number of channels in the descriptor, from 1 to 3, defaults to 3.
- **threshold** - same as for KAZE (threshold at which to accept a keypoint, defaults to 0.001).
- **nOctaves** - same as for KAZE (maximum octave evolution of the image, defaults to 4).
- **nOctaveLayers** - same as for KAZE (number of sublevels per scale level, defaults to 4).
- **diffusivity** - same as for KAZE, even uses KAZE constants (diffusivity type, default is DIFF_PM_G2).

The following algorithm parameter ranges were used for the parameter tuning:

- **descriptor_type** - as this parameter relates to the descriptor, default value is used.
- **descriptor_size** - as this parameter relates to the descriptor, default value is used.
- **descriptor_channels** - as this parameter relates to the descriptor, default value is used.

[24] https://docs.opencv.org/4.8.0/d8/d30/classcv_1_1AKAZE.html

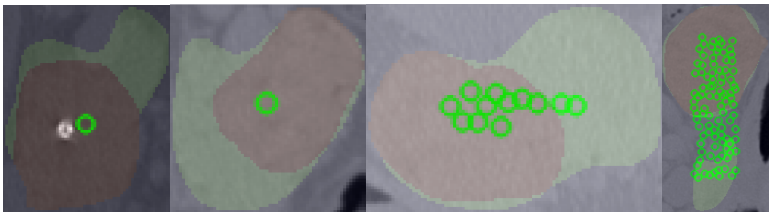
- **threshold** - range from 0 to 0.002 will be used.
- **nOctaves** - range from 1 to 10 will be used.
- **nOctaveLayers** - range from 1 to 10 will be used.
- **diffusivity** - all variations will be used.

Running the parameter tuning for the algorithm takes approximately 1 minute, which is indeed much faster then KAZE, and compared to all other algorithms is general, though performance in this specific task does not look good. Best achieved geometric mean is 0.427, highest balanced accuracy in top 10 results - 0.566. Best result also has TP=39, TN=183, FP=15, FN=159, sensitivity=0.197, specificity=0.924, with keypoints per image ranging from 0 to 103, 4.621 on average. The fraction of keypoints in the tumor zone is 0.768. Specificity is much higher than sensitivity, and this reveals that balanced accuracy is not a good metric in this case, as it becomes high though the algorithm actually performs not very well.

| Rank | nOctaveLayers | nOctaves | diffusivity | threshold | g-mean | b-acc | avg kp |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 1 | 2 | 0.000 | 0.427 | 0.561 | 4.621 |
| 2 | 6 | 4 | 3 | 0.000 | 0.427 | 0.561 | 3.747 |
| 3 | 5 | 3 | 3 | 0.000 | 0.427 | 0.561 | 4.515 |
| 4 | 9 | 1 | 0 | 0.000 | 0.427 | 0.561 | 4.657 |
| 5 | 9 | 1 | 0 | 0.000 | 0.427 | 0.561 | 4.444 |
| 6 | 6 | 5 | 3 | 0.000 | 0.427 | 0.561 | 3.687 |
| 7 | 10 | 1 | 3 | 0.000 | 0.425 | 0.566 | 2.914 |
| 8 | 6 | 2 | 3 | 0.000 | 0.425 | 0.566 | 3.278 |
| 9 | 5 | 4 | 3 | 0.000 | 0.425 | 0.566 | 3.111 |
| 10 | 6 | 5 | 1 | 0.000 | 0.425 | 0.566 | 2.874 |

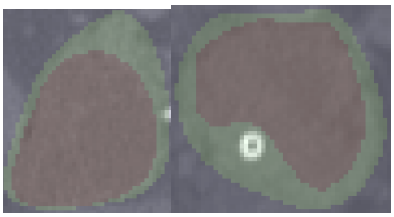
***8 table.*** *Top 10 Best Trials for AKAZE*

Table 8 table. provides a sample of best results in AKAZE parameter tuning. It seems that the threshold parameter has the biggest impact on the keypoints created by the algorithm (it has very small values). All other parameters can be set to any value in the selected range, and the results will be pretty similar.



**(a)** **(b)** **(c)** **(d)**

***19 figure.*** *AKAZE keypoint examples. a) and b) - with keypoints inside tumor region and no points outside of it; c) and d) with keypoints in both tumor region and outside of it*



(a) (b)

***20 figure.*** *AKAZE keypoint examples. a) and b) - no detected keypoints*

Figures 19 figure. and 20 figure. provide examples of AKAZE keypoints found in the images. The count of keypoints in each image varies significantly. Some images have a big count of keypoints, located in a specific region, but a lot of images have no keypoints at all. Just a small number of images have one or more keypoints inside the tumor region, and no keypoints outside of it. The algorithm performs poorly both in general and compared to other algorithms. It is worse than KAZE, which is a similar algorithm.

### 3.4.8 AGAST

OpenCV implements AGAST[25].

- **threshold** - defines the threshold for the keypoint detection, default is 10.
- **nonmaxSuppression** - defines whether to use nonmaximal suppression, default is True.
- **type** - defines the type of the detector, which can be one of AGAST_5_8, AGAST_7_12d, AGAST_7_12s or OAST_9_16, default is OAST_9_16.
- **threshold** - range from 0 to 100 will be used.
- **nonmaxSuppression** - as it can be either True or False, bot values will be used for the parameter tuning.
- **type** - all types will be used for the parameter tuning.

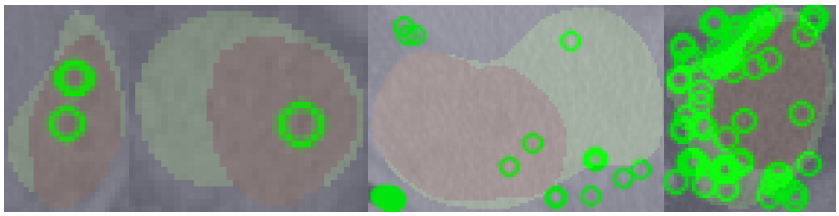
There are just a few parameters, and running the parameter tuning for the algorithm takes under a minute, so this is another fast algorithm. Best achieved geometric mean is 0.332, highest balanced accuracy in top 10 results - 0.439. Best result also has TP=120, TN=36, FP=162, FN=78, sensitivity=0.606, specificity=0.182, with keypoints per image ranging from 0 to 299, 26.455 on average. The fraction of keypoints in the tumor zone is 0.222. Both sensitivity and specificity are not high, with specificity being very low. The algorithm has the worst results compared to all other algorithms.

Table 9 table. provides a sample of best results in AGAST parameter tuning. Best results are achieved with threshold values from 5 to 10, and with type 0, which corresponds to constant AGAST_5_8. Algorithm does not perform well, compared to all other algorithms.

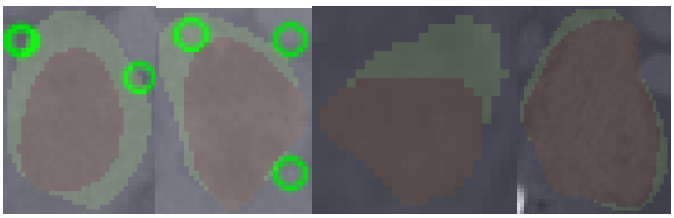
[25] https://docs.opencv.org/4.8.0/d7/d19/classcv_1_1AgastFeatureDetector.html

| Rank | type | nonmaxSuppression | threshold | g-mean | b-acc | avg kp |
|---|---|---|---|---|---|---|
| 1 | 1 | False | 7 | 0.332 | 0.394 | 26.455 |
| 2 | 1 | True | 7 | 0.321 | 0.374 | 13.318 |
| 3 | 0 | False | 6 | 0.320 | 0.336 | 12.894 |
| 4 | 1 | False | 8 | 0.316 | 0.338 | 18.237 |
| 5 | 1 | False | 9 | 0.313 | 0.313 | 13.308 |
| 6 | 1 | True | 8 | 0.311 | 0.328 | 8.394 |
| 7 | 1 | False | 10 | 0.309 | 0.316 | 10.227 |
| 8 | 1 | False | 6 | 0.308 | 0.439 | 40.768 |
| 9 | 2 | False | 8 | 0.304 | 0.333 | 19.192 |
| 10 | 1 | True | 6 | 0.300 | 0.419 | 21.626 |

***9 table.*** *Top 10 Best Trials for AGAST*




***21 figure.*** *AGAST keypoint examples. a) and b) - with keypoints inside tumor region and no points outside of it; c) and d) with keypoints in both tumor region and outside of it*




***22 figure.*** *AGAST keypoint examples. a) and b) - with keypoints outside of tumor region and not inside of it; c) and d) no detected keypoints*

Figures 21 figure. and 22 figure. provide examples of AGAST keypoints in the images. There are just 2 images that have keypoints in the tumor region, but not outside of it. There are also a lot of images that have either no keypoints, or a lot of keypoints, all of which are outside of the tumor region.

# Results and Conclusions

Keypoint detection and description algorithms are extensively used in medical image analysis. Though their primary purpose is image registration and matching, they are also sometimes used for other tasks, such as image classification and segmentation, usually in combination with machine learning and deep learning methods.

In this work, it was attempted to take a closer look into how keypoint detection algorithms are able to detect keypoints in pancreatic cancer CT images. The algorithms often perform better than chance, but the results are far from optimal. Out of the 7 algorithms tested, SIFT, SURF, ORB, BRISK and KAZE showed promising results in detecting keypoints inside the pancreatic tumor region. With specific parameters these algorithms achieved geometric mean scores of 0.687, 0.700, 0.715, 0.726 and 0.636 respectively. Balanced accuracy scores for these algorithms are 0.689, 0.702, 0.717, 0.727 and 0.636. As these algorithms are not classification or segmentation algorithms, it should not be expected of them to perform as well as machine learning or deep learning methods. However, as previous research has shown, they improve the results of these methods, which means that they are able to detect important features in the images of panreatic cancer. Hence trying to evaluate them in a standalone way could shed more light on how and why they improve the results of machine learning methods, and which keypoints detection algorithms could potentially be more useful in medical image classification and segmentation.

Some of the algorithms studied in this work, FAST, AKAZE and AGAST, did not perform well, with geometric mean scores of 0.475, 0.427 and 0.332 respectively (balanced accuracy - 0.477, 0.561 and 0.394). For FAST this can be explained by the fact that this algorithm is essentially a corner detector, and it uses simple but fast calculations to detect features, which might not be appropriate for medical images with usually low contrast and smooth features. Detection of blobs instead of corners is more appropriate here, as is evident by the results of SIFT, SURF, ORB, BRISK and KAZE, which all work differently than FAST or have some important modifications. For AKAZE, results are unexpected, as it is very similar to KAZE, and even claims to be more efficient, but in this specific task it performs worse than KAZE. For AGAST there were no studies found that would show its performance in medical image analysis, the reason for this might be its very poor performance in this type of tasks. It is very similar to FAST, but uses decision trees to select keypoints.

The algorithms that perform well in this task all detect keypoints in scale-space, not in one single layer like FAST and AGAST do. They also are able to detect blobs, not only corners which are not the type of features commonly present in medical images. Best results are achieved with BRISK and ORB, which both create multiple layers of scale space by downsampling original images, and then detect keypoints in these layers using FAST or AGAST. SIFT and SURF seem to perform a bit worse, and they use Gaussian blur for creating scale space, and Difference of Gaussian instead of FAST or AGAST. KAZE also differs by using non-linear diffusion for creating scale space, which in theory should work the best with medical images, but did not perform as well as BRISK or ORB in the case of pancreatic cancer CT images.

One of the biggest challenges was to find an evaluation method that would be appropriate

for this task. Keypoint detection algorithms are usually evaluated in the context of image matching, or image registration, where the goal is to find the same keypoints in different images of the same object. These evaluation methods do not suit the task, therefore an attempt was made to evaluate the algorithms in a way that would be similar to how they are used in medical image analysis for cancer detection, i. e. by approaching the task as a classification problem, and attempting to see if keypoints are detected inside the tumor region. Then there was another problem - when the region of the tumor is very small and does not cover approximately half of the image, the probablity of aa arbitraty keypoint to be inside the tumor is not 0.5 as we would expect in a classification task with two classes, but is much lower. Therefore all the evaluation metrics like accuracy or geometric mean have lower baseline and it appears that the algorithms are unable to yeld any meaningful results, i. e. results better than a chance, which is usually considered to be 0.5. But if in the images the tumor region covers more than half of the image, the baseline results start to be more than 0.5. Therefore the images in this study all have tumor zone covering approximately 50% of the image. This approach is not perfect, but it allows to compare the algorithms between each other, and give an insight on whether they are able to detect keypoints inside the tumor region better than a chance. To get more conclusive results, a different approach is needed, which would probably involve using keypoint detection algorithms in combination with machine learning methods for classification or segmentation tasks.

This research explored how keypoint detection algorithms perform in detecting keypoints in pancreatic cancer CT images, but these algorithms are not efficient as a standalone solution. It would be interesting to see how they perform in combination with machine learning and deep learning methods for classification and segmentation of pancreatic cancer. The existing studies analyzed in this thesis show that these algorithms can improve the results of classification significantly, but none of these studies were applied to pancreatic cancer, which present one of the biggest challenges in medical image analysis, both because of the high heterogeneity of the tumors, and outcomes of the disease. Additionally, application of these algorithms for pancreatic cancer image segmentation tasks could also be investigated.

# References and Sources


[ABD12] Pablo Fernández Alcantarilla, Adrien Bartoli, Andrew J. Davison. Kaze features. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pp. 214–227, Berlin, Heidelberg. Springer Berlin Heidelberg, 2012. ISBN: 978-3-642-33783-3.

[AD21] Elena G. Andrianova, LiliYa A. Demidova. An approach to image matching based on sift and orb algorithms. In *2021 3rd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA)*, pp. 534–539, 2021. https://doi.org/10.1109/SUMMA53307.2021.9632214. URL: https://ieeexplore.ieee.org/abstract/document/9632214.

[AHS+22] Roland Andersson, Caj Haglund, Hanna Seppänen, Daniel Ansari. Pancreatic cancer – the past, the present, and the future. *Scandinavian Journal of Gastroenterology*, 57(10):1169–1177, 2022. https://doi.org/10.1080/00365521.2022.2067786. eprint: https://doi.org/10.1080/00365521.2022.2067786. URL: https://doi.org/10.1080/00365521.2022.2067786. PMID: 35477331.

[AK17] Marwan Alshipli, Norlaili A. Kabir. Effect of slice thickness on image noise and diagnostic content of single-source-dual energy computed tomography. *Journal of Physics: Conference Series*, 851(1):012005, 2017-05. https://doi.org/10.1088/1742-6596/851/1/012005. URL: https://dx.doi.org/10.1088/1742-6596/851/1/012005.

[AK20] Mohammed Sahib Mahdi Altaei, Sura Yarub Kamil. Brain tumor detection and classification using SIFT in MRI images. *AIP Conference Proceedings*, 2292(1), 2020-10. ISSN: 0094-243X. https://doi.org/10.1063/5.0031014. eprint: https://pubs.aip.org/aip/acp/article-pdf/doi/10.1063/5.0031014/13596247/030004_1_online.pdf. URL: https://doi.org/10.1063/5.0031014. 030004.

[ARS19] Neneng Nur Amalina, Kurniawan Nur Ramadhani, Febryanti Sthevanie. Nuclei detection and classification system based on speeded up robust feature (surf). *EMITTER International Journal of Engineering Technology*, 7(1):1–13, 2019-06. https://doi.org/10.24003/emitter.v7i1.288. URL: https://emitter.pens.ac.id/index.php/emitter/article/view/288.

[AS22] Rif' Atunnisa, Raden Sumiharto. Pneumonia detection using extraction features oriented fast and rotated brief and convolutional neural network. *International Research Journal of Advanced Engineering and Science*, 7(2455-9024):103–107, 2022. URL: http://irjaes.com/wp-content/uploads/2022/10/IRJAES-V7N3P343Y22.pdf.

[ASY+19] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, Masanori Koyama. Optuna: a next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[ASK+20] Nagwa M. Aboelenein, Piao Songhao, Anis Koubaa, Alam Noor, Ahmed Afifi. Httu-net: hybrid two track u-net for automatic brain tumor segmentation. *IEEE Access*, 8:101406–101415, 2020. https://doi.org/10.1109/ACCESS.2020.2998601.

[BHP+97] W D Jr Bidgood, S C Horii, F W Prior, D E Van Syckle. Understanding and using dicom, the data interchange standard for biomedical imaging. eng. *J Am Med Inform Assoc*, 4(3):199–212, 1997-05. ISSN: 1067-5027 (Print); 1527-974X (Electronic); 1067-5027 (Linking). https://doi.org/10.1136/jamia.1997.0040199.

[BTV06] Herbert Bay, Tinne Tuytelaars, Luc Van Gool. Surf: speeded up robust features. In Aleš Leonardis, Horst Bischof, Axel Pinz, editors, *Computer Vision – ECCV 2006*, pp. 404–417, Berlin, Heidelberg. Springer Berlin Heidelberg, 2006. ISBN: 978-3-540-33833-8.

[CKS20] Swetha Parvatha Reddy Chandrasekhara, Mohan Govindsa Kabadi, Srivinay. A novel sift-svm approach for prostate cancer detection. *Journal of Computer Science*, 16(12):1742–1752, 2020-12. https://doi.org/10.3844/jcssp.2020.1742.1752. URL: https://thescipub.com/abstract/jcssp.2020.1742.1752.

[DCC+15] M. Ducreux, A. Sa. Cuhna, C. Caramella, A. Hollebecque, et al. Cancer of the pancreas: esmo clinical practice guidelines for diagnosis, treatment and follow-up. *Annals of Oncology*, 26:v56–v68, 2015. https://doi.org/10.1093/annonc/mdv295. URL: https://doi.org/10.1093/annonc/mdv295.

[DP23] Aman Dureja, Payal Pahwa. Integrating cnn along with fast descriptor for accurate retrieval of medical images with reduced error probability. *Multimedia Tools and Applications*, 82(12):17659–17686, 2023. https://doi.org/10.1007/s11042-022-13991-w. URL: https://doi.org/10.1007/s11042-022-13991-w.

[DRS15] Jurij Dolenšek, Marjan Slak Rupnik, Andraž Stožer. Structural similarities and differences between the human and the mouse pancreas. *Islets*, 7(1):e1024405, 2015.

[FAL+05] Megan Dann Fesinmeyer, Melissa A. Austin, Christopher I. Li, Anneclaire J. De Roos, Deborah J. Bowen. Differences in Survival by Histologic Type of Pancreatic Cancer. *Cancer Epidemiology, Biomarkers and Prevention*, 14(7):1766–1773, 2005-07. ISSN: 1055-9965. https://doi.org/10.1158/1055-9965.EPI-05-0120. eprint: https://aacrjournals.org/cebp/article-pdf/14/7/1766/1749242/1766-1773.pdf. URL: https://doi.org/10.1158/1055-9965.EPI-05-0120.

[Fer13] Pablo Fernández Alcantarilla. Fast explicit diffusion for accelerated features in nonlinear scale spaces. In 2013-09. https://doi.org/10.5244/C.27.13.

[HQK+19] Amy Harwelland, Devon Quick, Joel Kaufmann, Jon Runyeon, et al. *Anatomy and Physiology*. Oregon State University, 2019. URL: https://open.oregonstate.education/aandp/.

[HSA+19] Lal Hussain, Sharjil Saeed, Imtiaz A. Awan, Adnan Idris, Malik Sajjad Ahmed Nadeem, Qurat-ul-Ain Chaudhry. Detecting brain tumor using machines learning techniques based on different features extracting strategies. *Current Medical Imaging Reviews*, 15(6):595–606, 2019. ISSN: 1573-4056. https://doi.org/doi:10.2174/15734056146661807181235 33. URL: https://www.ingentaconnect.com/content/ben/cmir/2019/00000015/00000006/art00010.

[KAA22] Samuel Kusi-Duah, Obed Appiah, Peter Appiahene. Performance evaluation of state-of-the-art texture feature extraction techniques on medical imagery tasks. *Available at SSRN 4315803*, 2022.

[KDH+16] Muhammad Kashif, Thomas M. Deserno, Daniel Haak, Stephan Jonas. Feature description with sift, surf, brief, brisk, or freak? a general question answered for bone age assessment. *Computers in Biology and Medicine*, 68:67–75, 2016. ISSN: 0010-4825. https://doi.org/https://doi.org/10.1016/j.compbiomed.2015.11.006. URL: https://www.sciencedirect.com/science/article/pii/S0010482515003741.

[KDK+22] Povilas Kavaliauskas, Audrius Dulskas, Inga Kildusiene, Rokas Arlauskas, Rimantas Stukas, Giedre Smailyte. Trends in pancreatic cancer incidence and mortality in lithuania, 1998-2015. eng. *Int J Environ Res Public Health*, 19(2), 2022-01-15. ISSN: 1660-4601 (Electronic); 1661-7827 (Print); 1660-4601 (Linking). https://doi.org/10.3390/ijerph19020949.

[KJL+21] Yoo Jung Kim, Hyungjoon Jang, Kyoungbun Lee, Seongkeun Park, et al. Paip 2019: liver cancer segmentation challenge. *Medical Image Analysis*, 67:101854, 2021. ISSN: 1361-8415. https://doi.org/https://doi.org/10.1016/j.media.2020.101854. URL: https://www.sciencedirect.com/science/article/pii/S1361841520302188.

[LCS11] Stefan Leutenegger, Margarita Chli, Roland Y. Siegwart. Brisk: binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, pp. 2548–2555, 2011. https://doi.org/10.1109/ICCV.2011.6126542.

[Liu18] Yi Liu. Research status and prospect for ct imaging. In Morteza Sasani Ghamsari, editor, *State of the Art in Nano-bioimaging*, chap. 5. IntechOpen, Rijeka, 2018. https://doi.org/10.5772/intechopen.73032. URL: https://doi.org/10.5772/intechopen.73032.

[LL14] Eun Sun Lee, Jeong Min Lee. Imaging diagnosis of pancreatic cancer: a state-of-the-art review. eng. *World J Gastroenterol*, 20(24):7864–7877, 2014-06. ISSN: 2219-2840 (Electronic); 1007-9327 (Print); 1007-9327 (Linking). https://doi.org/10.3748/wjg.v20.i24.7864.

[LMA+16] Xiangrui Li, Paul S. Morgan, John Ashburner, Jolinda Smith, Christopher Rorden. The first step for neuroimaging data analysis: dicom to nifti conversion. *Journal of Neuroscience Methods*, 264:47–56, 2016. ISSN: 0165-0270. https://doi.org/https://doi.org/

10.1016/j.jneumeth.2016.03.001. URL: https://www.sciencedirect.com/science/article/pii/S0165027016300073.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. https://doi.org/10.1023/B:VISI.0000029664.99615.94. URL: https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[LQL+15] Wei Li, Yu Qian, Martin Loomes, Xiaohong Gao. The application of kaze features to the classification echocardiogram videos. In Henning Müller, Oscar Alfonso Jimenez del Toro, Allan Hanbury, Georg Langs, Antonio Foncubierta Rodriguez, editors, *Multimodal Retrieval in the Medical Domain*, pp. 61–72, Cham. Springer International Publishing, 2015. ISBN: 978-3-319-24471-6. URL: https://link.springer.com/chapter/10.1007/978-3-319-24471-6_6.

[MAA21] Kazi Safayet Md. Shabbir, Md. Imteaz Ahmed, Marzan Alam. Detection of glaucoma using orb (oriented fast and rotated brief) feature extraction. *Journal of Engineering Advancements*, 2(03):153–158, 2021. https://doi.org/10.38032/jea.2021.03.005. URL: http://scienpg.com/jea/index.php/jea/article/view/113.

[MHB+10] Elmar Mair, Gregory D. Hager, Darius Burschka, Michael Suppa, Gerhard Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In Kostas Daniilidis, Petros Maragos, Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pp. 183–196, Berlin, Heidelberg. Springer Berlin Heidelberg, 2010. ISBN: 978-3-642-15552-9. URL: https://doi.org/10.1007/978-3-642-15552-9_14.

[MKR+22] Tahereh Mahmoudi, Zahra Mousavi Kouzahkanan, Amir Reza Radmard, Raheleh Kafieh, Aneseh Salehnia, Amir H. Davarpanah, Hossein Arabalibeik, Alireza Ahmadian. Segmentation of pancreatic ductal adenocarcinoma (pdac) and surrounding vessels in ct images using deep convolutional neural networks and texture descriptors. *Scientific Reports*, 12(1):3092, 2022. https://doi.org/10.1038/s41598-022-07111-9. URL: https://doi.org/10.1038/s41598-022-07111-9.

[MMG+22] N. Jagan Mohan, R. Murugan, Tripti Goel, Parthapratim Roy. Fast and robust exudate detection in retinal fundus images using extreme learning machine autoencoders and modified kaze features. *Journal of Digital Imaging*, 35(3):496–513, 2022. ISSN: 1618-727X. https://doi.org/10.1007/s10278-022-00587-x. URL: https://doi.org/10.1007/s10278-022-00587-x.

[MPS+20] Michael Moore, Brandon Patterson, Sara Samuel, Helenmary Sheridan, Chris Sorensen. *Neuroimaging DICOM and NIfTI Data Curation Primer*. University of Minnesota. 2020. URL: https://hdl.handle.net/11299/216582.

[NT19] Viet Dung Nguyen, Thanh Hien Truong. Speeded-up robust feature descriptor for endochromoscopy images. In *2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, pp. 1–3, 2019. https://doi.org/10.1109/ITC-CSCC.2019.8793340.

[OCK+21] Kemal Oksuz, Baris Can Cam, Sinan Kalkan, Emre Akbas. One metric to measure them all: localisation recall precision (lrp) for evaluating visual detection tasks, 2021. arXiv: 2011.10772 [cs.CV].

[OHG23] Ghada Ouddai, Ines Hamdi, Henda Ben Ghézala. A comparative study of brisk, orb and daisy features for breast cancer classification. In *ICPRAM*, pp. 964–970, 2023. URL: https://www.scitepress.org/Papers/2023/119022/119022.pdf.

[PAK+21] Ingrida Pocė, Jaroslava Arsenjeva, Aistė Kielaitė-Gulla, Artūras Samuilis, Kęstutis Strupas, Gintautas Dzemyda. Pancreas segmentation in ct images: state of the art in clinical practice pancreas segmentation in ct images: state of the art in clinical practice. *Baltic journal of modern computing. Riga: University of Latvia.*, 9(1):25–34. 2021-09. URL: https://epublications.vu.lt/object/elaba:85416732/.

[PJA20] Neelakanta Rao Piratla, Pradeep Jakkula, ZOya Arfa. Enhanced computerized brain tumor detection with the features extracted from sift. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 889–894, 2020. https://doi.org/10.1109/ICICCS48265.2020.9121154.

[PWW+19] Yoga Dwi Pranata, Kuan-Chung Wang, Jia-Ching Wang, Irwansyah Idram, Jiing-Yih Lai, Jia-Wei Liu, I-Hui Hsieh. Deep learning and surf for automated classification and detection of calcaneus fractures in ct images. *Computer Methods and Programs in Biomedicine*, 171:27–37, 2019. ISSN: 0169-2607. https://doi.org/https://doi.org/10.1016/j.cmpb.2019.02.006. URL: https://www.sciencedirect.com/science/article/pii/S0169260718314652.

[RD06] Edward Rosten, Tom Drummond. Machine learning for high-speed corner detection. In Aleš Leonardis, Horst Bischof, Axel Pinz, editors, *Computer Vision – ECCV 2006*, pp. 430–443, Berlin, Heidelberg. Springer Berlin Heidelberg, 2006. ISBN: 978-3-540-33833-8.

[RRK+11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary Bradski. Orb: an efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pp. 2564–2571, 2011. https://doi.org/10.1109/ICCV.2011.6126544.

[SA10] Neeraj Sharma, Lalit M Aggarwal. Automated medical image segmentation techniques. eng. *J Med Phys*, 35(1):3–14, 2010-01. ISSN: 1998-3913 (Electronic); 0971-6203 (Print); 0971-6203 (Linking). https://doi.org/10.4103/0971-6203.58777.

[SAB+19] Amber L. Simpson, Michela Antonelli, Spyridon Bakas, Michel Bilello, et al. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. *CoRR*, abs/1902.09063, 2019. arXiv: 1902.09063. URL: http://arxiv.org/abs/1902.09063.

[SGG+18] Daniel Sanchez-Morillo, Jesús González, Marcial García-Rojo, Julio Ortega. Classification of breast cancer histopathological images using kaze features. In Ignacio Rojas, Francisco Ortuño, editors, *Bioinformatics and Biomedical Engineering*, pp. 276–286, Cham.

Springer International Publishing, 2018. ISBN: 978-3-319-78759-6. URL: https://doi.org/10.1007/978-3-319-78759-6_26.

[SMG+22] Seba Sahy, Sura Mahdi, Hassan Gheni, Israa Al-Barazanchi. Detection of the patient with covid-19 relying on ml technology and fast algorithms to extract the features. *Bulletin of Electrical Engineering and Informatics*, 11(5):2886–2894, 2022. ISSN: 2302-9285. https://doi.org/10.11591/eei.v11i5.4355. URL: https://beei.org/index.php/EEI/article/view/4355.

[SRK+23] B. A. Shamla, S. Ratheesha, S. Kalady, J. J. Chakola. Feature extraction based on orb-akaze for echocardiogram view classification. *International Journal of Electrical and Computer Engineering Systems*, 14(4):393–400, 2023. https://doi.org/10.32985/ijeces.14.4.3. URL: https://doi.org/10.32985/ijeces.14.4.3.

[TJA11] Pavan Tummala, Omer Junaidi, Banke Agarwal. Imaging of pancreatic cancer: an overview. eng. *J Gastrointest Oncol*, 2(3):168–174, 2011-09. ISSN: 2219-679X (Electronic); 2078-6891 (Print); 2078-6891 (Linking). https://doi.org/10.3978/j.issn.2078-6891.2011.036.

[UMS22] Mutammimul Ula, Muhathir, Ilham Sahputra. Optimization of multilayer perceptron hyperparameter in classifying pneumonia disease through x-ray images with speeded-up robust features extraction method. English. *International Journal of Advanced Computer Science and Applications*, 13(10), 2022. URL: https://www.proquest.com/scholarly-journals/optimization-multilayer-perceptron-hyperparameter/docview/2740752440/se-2. Copyright - © 2022. This work is licensed under https://creativecommons.org/licenses/by/4.0/ (the “License”). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2022-11-29.

[WZY+21] Shun Wang, Yan Zheng, Feng Yang, Le Zhu, et al. The molecular biology of pancreatic adenocarcinoma: translational challenges and clinical perspectives. *Signal Transduction and Targeted Therapy*, 6(1):249, 2021. https://doi.org/10.1038/s41392-021-00659-4. URL: https://doi.org/10.1038/s41392-021-00659-4.

[ZL20] ZhiYu Zhao, Wei Liu. Pancreatic cancer: a review of risk factors, diagnosis, and treatment. *Technology in Cancer Research & Treatment*, 19:1533033820962117, 2020. https://doi.org/10.1177/1533033820962117. eprint: https://doi.org/10.1177/1533033820962117. URL: https://doi.org/10.1177/1533033820962117. PMID: 33357065.