

TEKNOFEST
HAVACILIK, UZAY VE TEKNOLOJİ
FESTİVALİ

İNSANLIK YARARINA TEKNOLOJİ YARIŞMASI
PROJE DETAY RAPORU

PROJE KATEGORİSİ: Afet Yönetimi

PROJE ADI: Doğal Afetlerde Kullanılması Amaçlanan Su Altı Aracı Tasarımı

TAKIM ADI: AFSU

TAKIM ID: T3-22198-145

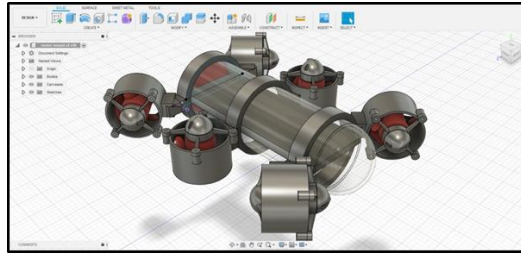
TAKIM SEVİYESİ: Lise

DANIŞMAN ADI: Hidayet KILCAN

1. Proje Özeti (Proje Tanımı):

Tsunami, sel ve taşkın gibi afetlerde can kaybının yaşanmasının en büyük nedeni bu kişilerin tespitinin oldukça zor olup yardımın geç yapılmasıdır. Projemiz afet durumlarında bu su altında kalmış kişilerin tespitini kolaylaştırmayı amaçlamaktadır. Afet zamanlarında su altında kalan vatandaşlarımızın can kaybının temel nedeni ilk yardımdaki gecikmedir. Bu gecikmenin nedenleri de afet anlarında su altının dalgıçlar için oldukça tehlikeli olup aynı zamanda görüş alanının oldukça kısıtlı olmasıdır. Ayrıca afet anında su altı dalgıçlar için de son derece tehlikelidir. Projemiz de bu problemin çözümü için bu tür arama kurtarma görevlerinde insansız su altı araçlarını kullanmayı planlamıştır. Küçük boyutlarından dolayı bu araçlar dalgıçlardan daha fazla alana girip daha çok bölgeyi araştırabilecektir. Bunu yapmayı planlarken ayrıca aracımızı değişik görüntü işleme teknikleri kullanarak otonom/yarı otonom hale getirip afetlerde çalışan arama kurtarma ekiplerinin daha efektif kullanılmasını sağlayacağız.

İnsansız su altı aracımızın yapımına ilk başta amaçlarımıza uygun bir su altı aracı gövdesi tasarlayarak başladık. Yaptığımız farklı tasarımlar arasında üretimi ve hareket kabiliyeti en uygun olan aşağıdaki tasarımı seçtik.



Tasarımımızda toplam 6 adet itici motor ve türbin bulunmaktadır. Bu türbinlerin ortadaki ikisi aracımızın batış ve tırmanış hareketini sağlayacaktır. Diğer 4 türbin ise aracın kendi ekseninde dönmesini sağlayacak aynı zamanda da yan yönlerde hareketini sağlayacaktır.

Türbinlere bağlı motorlar akrilik ana gövdedeki esler ve bir güç modülü aracılığıyla uçuş kartımıza (pixhawk) bağlanacaktır. Bu haliyle kara veya teknedeki bir operatör bir kumanda ve ya kontrol istasyonu ile aracı manuel olarak kullanabilecektir.

Aracımız herhangi bir kazazedeyi termal kamera sıcaklığının su altında kolayca dış ortamdan ayırt edilebileceğinden termal kameradan aldığımız verilerdeki rgb değerlerini belirleyeceğimiz eşik değerlerle karşılaştırıp olası kazazedeleri tespit edecektir. Tespit etmede kullanacağımız termal kamera aldığı görüntüyü Raspberry kartına gönderecek ve burada aldığımız termal görüntüdeki kırmızı veya sarı renkleri arayacaktır. Böylece muhtemel bir canlı tespit edilmiş olacaktır.

Tespit etme aşamasından sonra aracın hareketini su altındaki engellere takılmadan yapması için termal özelliği olmayan bir kamera kullanacağız. Bu kamera aracın üstünde bulunacak güçlü aydınlatma sistemleri sayesinde etrafındaki nesnelere

kenarlarını seçmemizi kolaylaştıracak. Raspberry'ye bağlı olacak bu kamera ilk başta aldığı görüntüyü bu karta gönderecek. Kartta ilk başta yapılacak iş aldığımız görüntüye Gaussian filtresini uygulamak olacak. Böylece görüntümüzün noise değeri düşüp kenar tespitine daha uygun hale gelecektir. Daha sonra gradient calculation işlemini yaparak elimizdeki bu görüntüdeki renk ve yoğunluk farkı en yüksek olan bölgeleri yani muhtemel kenarları seçecektir. Fakat bu işlemin ardından kenar olarak belirttiğimiz bu bölgeler üst üste binmiş veya karışık halde karşımıza çıkabilir. Bu problemi ortadan kaldırıp kenarları daha keskin hale getirmek için non max suppression adlı algoritmaya kenarları seçili görüntümüzü sokacağız. Üst üste binen bu kenarlardan kurtulunca elimizdeki görsele yüksek değerde bir threshold uygulayarak kenarların tamamen ortaya çıkmasını sağlayacağız. Bundan sonra yapacağımız son işlem ise diğerlerine nazaran çok zayıf olan kenarları başka bir algoritma ile kaldırmak olacaktır. Böylece elimizde kenarları keskin bir şekilde belirlenmiş bir görüntüye sahip olacağız. Kenarları belirledikten sonra bu nesnelere ile aramızdaki mesafeyi belirleyerek nereye hareket etmemiz gerektiğini hesaplamamız lazım. Bu amaçla da kendi gözünüz ile de deneyebileceğiniz bir yöntem kullanacağız. Gözünüzü önünüze bakacak bir şekilde sabitleyin ve kafanızı sağ veya sol doğrultusunda hareket ettirin. Önünüzdeki tüm cisimler gözünüzden eşit mesafe uzaklaşacak olsa bile siz size en uzak şeyi en az hareket etmiş gibi, en yakın şeyi de en fazla hareket etmiş gibi göreceksiniz. Bizim elimizde de bu nesnelere kenarlarının verisi var ve aracımızdaki kamera ile de aynı işlemi yaptığımızda bu kenarlar eşit oranda hareket etmiş olsalar da görüntümüze farklı oranda hareket etmiş olacaklardır. Bu kenarların hareket oranının hesabından da cisimlerin bize olan uzaklığını ölçerek gerekli hareket komutlarını Pixhawk uçuş kartına ileticek ve aracımızın otonom hareketini sağlayacağız.

2. Problem/Sorun:

Projemiz için belirlediğimiz problem afet zamanlarında su altında kalan insanların büyük bir yüzdesinin bu olay sonucu ölüm ile karşılaşmasıdır. Bu gibi afet zamanlarında bu problemin en büyük nedeni de kişi tespitinin hızlıca yapılamaması ve ilk yardımın zamanında gelememesidir. Ve başka bir problem olarak su altının kazazedelerin yanında dalgıçlar için de oldukça tehlikeli olabilmesidir. Biz de bu iki ana sorundan yola çıkarak dalgıçlarımızı bu tür tehlikeli işlemlerden uzak tutup acil durumdaki kişilerin ihtiyacına en hızlı nasıl yetişiriz sorularına cevap aradık.

3. Çözüm:

Projemizde hızlı bir şekilde arama ve kişilerin yerini tespit etme görevlerinde kullanılacak bir su altı aracı geliştireceğiz. Aracın insansız olmasından dolayı kazazedeleri tespit ederken dalgıçları riske atmayacağız. Ve dalgıçlardan daha küçük boyutlu olmaları sayesinde arama görevlerini daha etkin bir şekilde yerine getirecektir. Ayrıca üstünde çalıştığımız görüntü işleme teknikleri sayesinde aracımızın kıyı veya teknedeki bir operatör tarafından kullanılmasını oldukça kolaylaştıracak ve eğer amacımıza ulaşırsak aracın otonom/yarı otonom olarak kumanda edilmesi sağlanacaktır. Otonom çalışmasını olabildiğince az hata payı ile sağlayabilirsek araçların operatör sayısını azaltıp arama kurtarma ekiplerinin daha efektif çalışmasını

sağlayabiliriz. Çözüm olarak ürettiğimiz prototipimiz 6 türbinle hareketini sağlayan, iki adet kameraya sahip bir insansız su altı aracıdır.

4. Yöntem:

Bu problem doğrultusunda tasarladığımız su altı aracı 6 adet itici motoru ile tüm eksenlerinde hareket kabiliyetine sahip üstünde bir adet kamera, bir adet termal kamera ve güçlü bir aydınlatmaya sahip bir su altı aracıdır. Termal kamera olası kazazedelerin tespitinde kullanılacak iken normal kameramız da manuel kontrolde operatöre bilgi sağlayacak ve aynı zamanda üstünde hala çalışmakta olduğumuz görüntü işleme teknikleri sayesinde kendi etrafındaki nesnelere olabildiğince uzak durarak otonom şekilde çevresindeki kazazedeleri arayacaktır.

Su altı aracının manuel hareketi kısmında geçmişte yapmış olduğumuz araçlardaki tecrübemizden yola çıkarak bir sorun yaşamayacağımızı düşünüyoruz. Fakat görüntü işleme için ilk kez böylesine karmaşık bir ortamı seçtik. Şu ana kadar yaptığımız görüntü işleme işlemleri bize basit şekillerde (arka plandan kolayca ayrılabilen şekiller) iyi bir sonuç vermektedir fakat programımızın çalışmasında hız düşüşleri yaşamaktayız. Su altı çok daha karmaşık olacağından bu hız düşüşünün daha fazla olacağını öngörmekteyiz ve otonom hareket etme çalışmalarımızda önceliğimizi bu yaptığımız programın optimizasyonu üzerine yoğunlaştırdık. Bu problemi aracımızda kullandığımız raspberry kart yerine daha güçlü işlem gücüne sahip başka kartlar kullanarak da aşabiliriz.

5. Yenilikçi (İnovatif) Yönü:

Yaptığımız literatür taramasında birkaç farklı sualtı keşif robotuna rastladık fakat bizim projemizin aksine bunların hiçbiri afet anında çalışmaya elverişli değildir. Projemizde tasarladığımız su altı aracının tasarımı tamamen bize ait ve özgündür. Gövde parçalarının çoğunu Fusion 360 programında tasarlayıp 3D yazıcıdan baskısını aldık. Bunun yanında geliştirmekte olduğumuz görüntü işleme üzerine kurulu otonom hareket etme ve tespit etme sistemi ile eğer başarıya ulaşırsak arama kurtarmada insansız su altı araçlarının kullanımında büyük bir yeniliğe imza atılmış olacağız. Projemiz piyasada bulunan benzer ürünlerden dış tasarımı ve kullanılan bileşenler konusunda farklılık göstermektedir.

Aracımızın kullanım alanı afet anıyla sınırlı olmamaktadır. Özellikle yaz aylarında barajlara serinlemek için girip boğulan insanların haberlerini sıklıkla duymaktayız. Aracımız bu gibi durumlarda ceset aramak için de kullanılabilir. Ayrıca açık denizde keşif robotu olarak kullanılabilir.

6. Uygulanabilirlik:

Projemiz büyük çaplı olmayan destek ve yatırımlarla acil durum ekiplerine verilecek kısa bir eğitimle afet durumlarında kullanılmaya hazır hale getirilebilir. Projemiz hayata

geçirildiğinde ülkemizde AFAD olmak üzere afet müdahale ekiplerinin sel, taşkın vb. afetler için en temel ekipmanları arasında yer alacağını öngörmekteyiz.

Ayrıca projemizin başarısına bağlı olarak afet anlarında kullanılması amacıyla çoğu ülkeden müşteri çıkacağına inanıyoruz çünkü bu amaçla kullanılacak insansız su altı araçlarının sayıları oldukça az. Var olanların da ücreti oldukça yüksektir.

Projemizin riskli yanlarından bahsetmek gerekirse afet alanlarında hareket etme konusunda sıkıntı yaşamayacaktır fakat kazazedelerin tanınmasında her zaman bir hata payı olacaktır. Projemiz için desteği alıp aracımızı tamamladıktan sonra kendimizi daha farklı otonom tespit yöntemleri üzerinde geliştirip bu hata payının azaltılması üzerinde çalışacağız.

7. Tahmini Maliyet ve Proje Zaman Planlaması:

Projemizin afet durumlarında yeterince dayanıklılığa sahip, elektronik donanımı ihtiyaçlarımızı daha iyi karşılayacak bir versiyonu bize ortalama 12.900 TL fiyat bazında olacağını hesapladık. Fakat bu fiyatın büyük bir miktarını (6000 TL) termal kamera kapsamaktadır. Eğer yerli üretilen bir termal kamera modeli bize yeteri kadar verim verir ise veya bu aracın seri üretiminde bu kameraların toptan alımı söz konusu olursa bu maliyeti oldukça fazla düşecektir. Fakat bu fiyatın yüksek olması sizi şaşırtmamalı. Çünkü günümüzde piyasada satılan video çekimi amacıyla kullanılan insansız su altı araçlarının fiyatları 10-15 bin TL'den başlayıp 150 bin TL ye kadar çıkmaktadır. Prototipimizin üretiminde gereken parça listesi:

Ürün	Miktar	Adet Fiyatı
Lotus Makaralı Uzatma Kablo Seyyar Priz 3x1,5 35 Metre	1	95,00 TL
2Pin IP68 16mm Su Geçirmez Konnektör Takım	10	7,98 TL
LOGITECH C270 HD WEBCAM	1	253,00 TL
Pleksiglass Fanus	1	23,60 TL
Pleksiglass Tüp	1	200,60 TL
ABG Marka Beyaz ABS Filament 1.75 mm	5	89,00 TL
30 METRE CAT6 LAN RJ45 JACKLI İNTERNET MODEM KABLOSU	1	71,00 TL
FPV UBEC DEVRESİ 5V 3A JST KONNEKTÖR	1	24,90 TL
9000 lumen led çip	4	6,90 TL
4-25v-ayarli-voltajdusurucu	1	57,08 TL

Ürün	Miktar	Adet Fiyatı
Waterproof 3900KV Fırçasız Motor, 60A ESC Combo Seti	6	438,60 TL
Analog Su Basınç Sensörü	1	137,00 TL
Raspberry Pi 4 - 2GB RAM	1	350,99 TL
link-tl-poe200	1	173,00 TL
Jiroskop/ivmeölçer	1	8,44 TL
Adet Pixhawk 2.4.8 Uçuş Kontrol Kartı	1	2.118,81TL
Bosch Gtc 400 C termal kamera	1	5.964,24 TL
Logitech F310 Gamepad	1	188,61 TL
APM Pixhawk Power BEC Modülü 30V 90A	1	90,00 TL

TOPLAM: 12.918,07 TL

	Aralık	Ocak	Şubat	Mart	Nisan	Mayıs	Haziran	Temmuz	Ağustos	Eylül
Literatür Taraması	X	X	X	X	X	X	X			
Tasarım		X	X	X	X					
Yazılım				X	X	X	X			
3D Baskı					X	X	X	X		
Elektronik Sistem						X	X	X	X	
Sızdırmazlık Testleri								X	X	
Montaj							X	X	X	X
Final Testleri										X

Zaman Planlaması Tablosu

8. Proje Fikrinin Hedef Kitle (Kullanıcılar):

Proje fikri su altı arama ve kurtarma ekiplerine yardımcı olmayı hedeflemektedir. Dalgıçlar için tehlike arz eden sularda aracın devreye girmesi planlanmıştır.

9. Riskler:

Projenin hayata geçirilmesi durumunda araç suyun hızla sürüklediği farklı nesnelere çarpabilir fakat aracın tasarımında koruyucu bir dış kaplama yapılması halinde bu sorun büyük ölçüde çözümlenmiş olur. Görüntü işlemeyi kullanırken insan tanımlamada hata yapma olasılığı vardır. Bu sorunun önüne geçmek için insan tanımlamada termal kamera kullanılabilir. Bu maliyeti biraz artıracak olsa da daha kesin çözüm vereceğini düşünmekteyiz.

Proje Hedefleri/ Etki	ETKİ ARALIĞI				
	Çok Düşük 0,5	Düşük 0,1	Orta 0,2	Yüksek 0,4	Çok Yüksek 0,8
Maliyet	Maliyet bütçe tahminlerini tutar, küçük bütçesel kaymalar yaşanabilir.	Maliyet hedeflenen bütçeyi çok az aşar,	Maliyet hedeflenen bütçeyi %5-%20 arasında aşar.	Maliyet hedeflenen bütçeyi %20-%50 arasında aşar.	Maliyet bütçe tahminleri doğrultusunda karşılanamayacak düzeye gelir.
Zaman	Zaman planlamasına etkisi göz ardı edilebilecek kaymalar yaşanabilir.	Zaman planlamasını çok fazla etkilemeyecek gecikme yaşanabilir.	Zaman planlamasını biraz etkileyecek 1-3 ay arası gecikme yaşanabilir.	Projenin hedefine ulaşma olasılığını azaltacak 3-4 ay arası gecikme yaşanabilir.	Projenin zamanında hedefine ulaşamayacağı kadar gecikme yaşanabilir.
Teknik	Projeye hiç etki etmeyecek veya çok az etki gösterecek arıza	Teknik performansı çok az etkileyecek arıza	Teknik performansı biraz etkileyecek ancak düzeltilmesi yapılabilecek arıza	Teknik performansı önemli ölçüde etkileyecek ve düzeltilmesi zor arıza	Hedeflere ulaşmasını engelleyecek düzeltilmez arıza

Risk Matrisi

10. Proje Ekibi:

Takım Lideri: Arda ÖZTÜRK

Adı Soyadı	Projedeki Görevi	Okul	Projeyle veya problemle ilgili tecrübesi
Arda ÖZTÜRK	Devre tasarımı, kodlama, raporlama	Dumlupınar Bilim ve Sanat Merkezi	2 yıldır benzer alanda çalışmakta
Zeynep Sezin TÜMER	Maket tasarımı, kodlama, raporlama	Dumlupınar Bilim ve Sanat Merkezi	4 yıldır benzer alanda çalışmakta

11. Kaynaklar:

Autonomous underwater vehicle. (2020, Şubat 26). Wikipedia:

https://en.wikipedia.org/wiki/Autonomous_underwater_vehicle adresinden alındı

Canny edge detector. (2020, Nisan 14). Wikipedia:

https://en.wikipedia.org/wiki/Canny_edge_detector adresinden alındı

Goswami, S. (tarih yok). *Reflections on Non Maximum Suppression (NMS)*. Medium:

<https://medium.com/@whatdhack/reflections-on-non-maximum-suppression-nms-d2fce148ef0a> adresinden alındı

Rescue Diver. (2020, Nisan 18). Wikipedia: https://en.wikipedia.org/wiki/Rescue_Diver adresinden alındı

Team, D. (2020, Ocak 7). *Project in Python – Colour Detection using Pandas & OpenCV*. Data Flair:

<https://data-flair.training/blogs/project-in-python-colour-detection/> adresinden alındı

Unmanned submarines seen as key to dominating the world's oceans. (2019, Ekim 15). Military Aerospace Electronics:

<https://www.militaryaerospace.com/unmanned/article/14068665/unmanned-underwater-vehicles-uuv-artificial-intelligence> adresinden alındı

EK 1

Termal kameranın görüntüsünün işlenmesi için olan kodlar.

```
# import the necessary packages
import numpy as np
import argparse
import cv2

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", help = "path to the image")
args = vars(ap.parse_args())

# load the image
image = cv2.imread(args["image"])

# define the list of boundaries
boundaries = [
    ([17, 15, 100], [50, 56, 200]),
    ([86, 31, 4], [220, 88, 50]),
    ([25, 146, 190], [62, 174, 250]),
    ([103, 86, 65], [145, 133, 128])
]
```

```
# loop over the boundaries
for (lower, upper) in boundaries:
    # create NumPy arrays from the boundaries
    lower = np.array(lower, dtype = "uint8")
    upper = np.array(upper, dtype = "uint8")

    # find the colors within the specified boundaries and apply
    # the mask
    mask = cv2.inRange(image, lower, upper)
    output = cv2.bitwise_and(image, image, mask = mask)

# show the images
cv2.imshow("images", np.hstack([image, output]))
cv2.waitKey(0)
```

Otonom harekette kullanılan görüntüye Gaussian filtresi ve gradient calculation işlemlerinin uygulanması.

```
import numpy as np

def gaussian_kernel(size, sigma=1):
    size = int(size) // 2

    x, y = np.mgrid[-size:size-1, -size:size-1]
    normal = 1 / (2.0 * np.pi * sigma**2)
    g = np.exp(-((x**2 + y**2) / (2.0*sigma**2))) * normal
    return g

from scipy import ndimage

def sobel_filters(img):
    Kx = np.array([[[-1, 0, 1], [-2, 0, 2], [-1, 0, 1]], np.float32])
    Ky = np.array([[1, 2, 1], [0, 0, 0], [-1, -2, -1]], np.float32)

    Ix = ndimage.filters.convolve(img, Kx)
    Iy = ndimage.filters.convolve(img, Ky)

    G = np.hypot(Ix, Iy)
    G = G / G.max() * 255
    theta = np.arctan2(Iy, Ix)

    return (G, theta)
```

Görüntümüze non max suppression algoritmasının uygulanması için kodlar.

```
def non_max_suppression(img, D):
    M, N = img.shape
    Z = np.zeros((M,N), dtype=np.int32)
    angle = D * 180. / np.pi
    angle[angle < 0] += 180

    for i in range(1,M-1):
        for j in range(1,N-1):
            try:
                q = 255
                r = 255

                #angle 0
                if (0 <= angle[i,j] < 22.5) or (157.5 <= angle[i,j] <= 180):
                    q = img[i, j+1]
                    r = img[i, j-1]

                #angle 45
                elif (22.5 <= angle[i,j] < 67.5):
                    q = img[i+1, j-1]
                    r = img[i-1, j+1]

                #angle 90
                elif (67.5 <= angle[i,j] < 112.5):
                    q = img[i+1, j]
                    r = img[i-1, j]

                #angle 135
                elif (112.5 <= angle[i,j] < 157.5):
                    q = img[i-1, j-1]
                    r = img[i+1, j+1]

                if (img[i,j] >= q) and (img[i,j] >= r):
                    Z[i,j] = img[i,j]
                else:
                    Z[i,j] = 0

            except IndexError as e:
                pass

    return Z
```

Görüntümüze yüksek seviyede threshold uygulamamız için gerekli kodlar.

```
def threshold(img, lowThresholdRatio=0.05, highThresholdRatio=0.09):
    strong_i, strong_j = np.where(img >= highThreshold)
    zeros_i, zeros_j = np.where(img < lowThreshold)

    highThreshold = img.max() * highThresholdRatio;
    lowThreshold = highThreshold * lowThresholdRatio;

    weak_i, weak_j = np.where((img <= highThreshold) & (img >= lowThreshold))

    M, N = img.shape
    res = np.zeros((M,N), dtype=np.int32)

    res[strong_i, strong_j] = strong
    res[weak_i, weak_j] = weak

    weak = np.int32(25)
    strong = np.int32(255)

    return (res, weak, strong)
```

Görüntümüzdeki zayıf kenarları ortadan kaldırmamız için gerekli kodlar.

```
def hysteresis(img, weak, strong=255):
    M, N = img.shape
    for i in range(1, M-1):
        for j in range(1, N-1):
            if (img[i,j] == weak):
                try:
                    if ((img[i+1, j-1] == strong) or (img[i+1, j] == strong) or (img[i+1, j+1] == strong)
                        or (img[i, j-1] == strong) or (img[i, j+1] == strong)
                        or (img[i-1, j-1] == strong) or (img[i-1, j] == strong) or (img[i-1, j+1] == strong)):
                        img[i, j] = strong
                    else:
                        img[i, j] = 0
                except IndexError as e:
                    pass
    return img
```

3d baskısını aldığımız gövde bileşenlerimiz ve kodlarımızı denemek için yaptığımız devre.

