

TEKNOFEST İSTANBUL
HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

YAPAY ZEKA YARIŞMASI
ÖN TASARIM RAPORU

TAKIM ADI: Graduate Student Descent
TAKIM ÜYELERİ: Dorukhan Afacan, Enis Simsar, Ömer
Kırbıyık, Hüsnü Şensoy

İçindekiler

1	<i>Takım Şeması.....</i>	3
2	<i>Kullanılması planlanan algoritmalar ve yazılım mimarisi.....</i>	3
2.1	Güncel Bilimsel Yöntemler	3
2.1.1	Mimariler	3
2.2	Yazılım Mimarisi	4
2.2.1	Sonuç Çıkarma (Inference Mimarisi)	4
3	<i>Kullanılması planlanan donanımlar ve özellikleri.....</i>	5
3.1	Eğitim Donanımları	5
3.2	Test Donanımları	5
4	<i>Özgünlük</i>	5
4.1	Model	5
5	<i>Proje Takvimi.....</i>	5

1 Takım Şeması

Adı Soyadı	Temel Görev
Dorukhan Afacan	<ul style="list-style-type: none"> Ekip koordinasyonunun sağlanması ve pipeline oluşturulması
Enis Simsar	<ul style="list-style-type: none"> Pipeline oluşturulması ve modellerin testi
Ömer Kırbıyık	<ul style="list-style-type: none"> Pipeline oluşturulması ve modellerin testi
Hüsnü Şensoy	<ul style="list-style-type: none"> Container mimarisine uygun altyapı hazırlığı

2 Kullanılması planlanan algoritmalar ve yazılım mimarisi

2.1 Güncel Bilimsel Yöntemler

2.1.1 Mimariler

2.1.1.1 Çift Aşamalı Mimariler (RCNN Tabanlı Mimariler)

RCNN: Güncel nesne tanıma mimarilerinin temelini oluşturan yapı *region proposal* ve nesne tanıma modüllerinden oluşur. *Selective search*¹ ile önerilen bölgelerden *feature* vektörü CNN ile elde edilir. Sonrasında SVM ile nesnenin sınıfı ve *bounding box*'i belirlenir.

Fast RCNN² & Faster RCNN³: Fast RCNN, imajı önce CNN ile elde edilen *feature* vektör üzerinden bölge önerimi mekanizmasını çalıştırır. Faster RCNN ise *selective search* yerine *Region Proposal Network (RPN)* kullanarak bölge önerimi kısmını da sinir ağları ile kurmuştur.

R-FCN⁴: Faster RCNN'deki *region proposal* yeniden boyutlandırılarak hızlandırılmıştır.

2.1.1.2 Tek Aşamalı Mimariler

YOLO⁵ & SSD⁶: Gerçek zamanlı nesne tanıma imkanı veren YOLO, *bounding box (BB)* ve sınıf tahminlerini tek bir sinir ağı içinde bir geçişte yapmakta ve gerçek-zamanlıya yakın tahminler üretmektedir. SSD, YOLO gibi belirlenmiş sayıda *BB* için nesne içermeye ihtimalini skorlayan bu yapıda, YOLO'nun başarısını artırmak için daha fazla CNN birimi ve *skip connectionlar* kullanılmıştır.

¹ Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, 104(2), 154-171.

² Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91-99).

³ Dai, J., Li, Y., He, K., & Sun, J. (2016). R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems* (pp. 379-387).

⁴ Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

⁵ Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.

⁶ Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988).

RetinaNet⁷: Farklı ölçekteki nesneler için daha iyi bir model olan *Feature Pyramid Network* öne sürülmüştür. Modelin getirdiği yenilik olan *focal loss* $FL(p_t) = -(1 - p_t)^{\gamma} \log(p_t)$ şeklindedir. Kolay örneklerin ağırlığı azaltılarak zor örneklerin fazla katkıda bulunması amaçlanmıştır.

2.1.1.3 Mimarilerin Karşılaştırılması

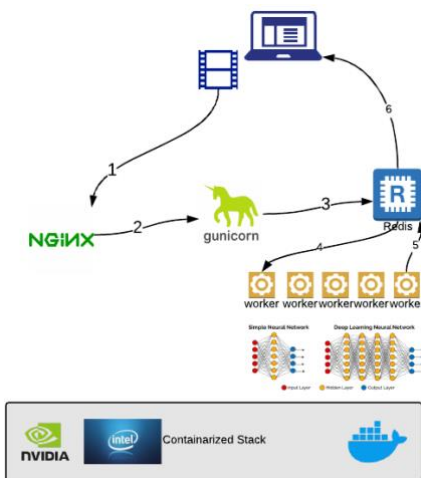
Mimari İsmi	Backbone	VOC 12	FPS	Notlar
RCNN	AlexNet	58.5	< 0.1	Eğitim hız ve hafıza açısından zayıf.
Fast RCNN	Alexnet, VGG	68.4	< 1	Uçtan uca sisteme sahip, eğitim yavaş
Faster RCNN	VGG	70.4	< 5	Hızlı <i>Region Proposal Network</i> yapısı, test real time'a göre yavaş
R-FCN	Resnet101	77.6	< 10	Tamamen CNN tabanlı, görece hızlı
YOLO	GoogLeNet	57.9	< 25	Çok hızlı, başarımı daha düşük
SSD	VGG	74.9	< 60	Hız ve başarımları daha dengeli
RetinaNet	FPN	?	< 8	Başarımları yüksek, küçük nesneleri tanımda başarılı

2.2 Yazılım Mimarisi

Geliştirmelerin tamamı Docker'da çalışan Python platformlarında yapılmaktadır. Bu sayede, farklı bulut veya donanım ortamları arasındaki farklılıklardan asgari ölçüde etkilenilmektedir. Ayrıca Python kütüphaneleri arasındaki uyum sorununun önüne geçilmektedir.

2.2.1 Sonuç Çıkarma (Inference Mimarisi)

Sonuç çıkarma (inference) mimarisi diyagramdaki gibi çalışmaktadır:



1. İstemci, elinde bulunan video akışını NGINX'e göndermeye başlar (STREAMING)
2. Gunicorn, videoyu alt kırımlara (segment) bölerek Redis (3) kuyruğuna kırılan kısmın detaylarıyla (başlangıç bitiş zamanları, video anahtarı vb.) depolar.
3. Modeli içeren bir çıkarım havuzu, 2. aşamada kuyruğa gönderilen segmentleri alır ve obje aramasını yapar. Daha sonra çıktıları bir başka Redis kuyruğuna gerekli detaylarla (alt segment anahtarı, obje detayları, vb.) depolar.

⁷ Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255). Ieee

4. Bu sayede istemci bloklanmadan, videoyu göndermeye devam ederken sonuçları toplar.

3 Kullanılması planlanan donanımlar ve özellikleri

3.1 Eğitim Donanımları

1. Geliştiricilerin iş istasyonları, modellerin geliştirme aşamasında kullanılan birimlerdir. Intel 4 core @ 3.7 Ghz, 32 GB memory ve NVIDIA GeForce® GTX 1080 ihtiva eder.
2. Global Maksimum Lab ortamı, artan veride kaynak sıkıntısını çözmek için bir havuz ortam olan küme kullanılmaktadır. 4 adet 24 core @ 2.6 Ghz – 256 GB memory sunucu barındırır.

3.2 Test Donanımları

Test donanımı olarak uzmanlarımızın kullandığı iş istasyonları kullanılacaktır.

4 Özgünlük

4.1 Model

Problem özelinde çok sayıda küçük boyutta nesnelerin olması beklenmektedir. Yeni bir yöntem olan FSAF⁸ yapısı *feature pyramid* ağlarına eklenerek bu objelerin başarılı bir şekilde tespit edilmesi sağlanacaktır. FSAF yapısında *feature pyramid*'in her seviyesinde, anchor kullanmayan bir parça eklenerek objelerin dinamik bir şekilde en uygun *feature* seviyesine atanması sağlanacaktır.

5 Proje Takvimi

#	Traits	Title	Given Work	Given Earliest Start	Resources	Predecessors	Q1 / 2	Q2 / 2019	Q3 / 2019	Q4 / 2019	Q1						
				Start			03	04	05	06	07	08	09	10	11	12	01
0	📅	T3 Yapay Zekâ Yarışması		19 Mar 2019	Enis Simsar; Doruk...												
1	🕒	Ön Tasarım Raporunun Hazırlanması		19 Mar 2019	Enis Simsar; Dorukhan Afacan; Ömer Kirbiyi...												
2	🕒	Ön Tasarım Raporu Teslim		15.05.2019													
3	🕒	Ön Eleme Sonuçlarının Açıklanması		30.05.2019													
4	🕒	Verinin Teslim Alınması		30.05.2019													
5	🕒	Pipeline Kurgusu		15.05.2019	Enis Simsar; Doruk...	4											
6		Veri Preprocessing			Enis Simsar; Dorukh...												
7		Model oluşturma			Ömer Kirbiyik; Dor...	6											
8		Baseline model			Ömer Kirbiyik; Doru...												
9		Final model			Enis Simsar; Dorukh...												
10		Model eğitilmesi/en iyileme			Enis Simsar; Dorukh...	6											
11		Test verisi üzerinden IoU skor hesaplanması			Ömer Kirbiyik; Dorukhan Afacan;...												
12		Baseline IoU			Ömer Kirbiyik; Doru...	8											
13		Final IoU			Ömer Kirbiyik; Doru...	7											
14	🕒	Pipeline MVP		15.05.2019		6; 8; 12											
15	🕒	Demo Application Hazırlanması		15.05.2019	Hüsnü Şensoy; Enis Simsar; Dorukhan...												
16	🕒	Container mimarisinde ortamların kurulması		30.05.2019	Hüsnü Şensoy												
17		Modellerin container mimarisine devreye alınması			Enis Simsar; Dorukhan Afacan; Ömer Kirbiyi...	8; 16											
18	🕒	Demo MVP		14 Jun 2019		16; 17											
19		Final Demo Hazırlığı				18											
20	🕒	Soru Cevap		15 Jun 2019		14; 18											
21	🕒	Kritik Tasarım Raporunun Hazırlanması		16 Jun 2019	Hüsnü Şensoy; Ömer Kirbiyik; Dorukhan A...	20											
22	🕒	Kritik Tasarım Raporu		15 July 2019													
23	🕒	Sonuçların Açıklanması		30 July 2019													

⁸ Zhu, C., He, Y., & Savvides, M. (2019). Feature Selective Anchor-Free Module for Single-Shot Object Detection. arXiv preprint arXiv:1903.00621