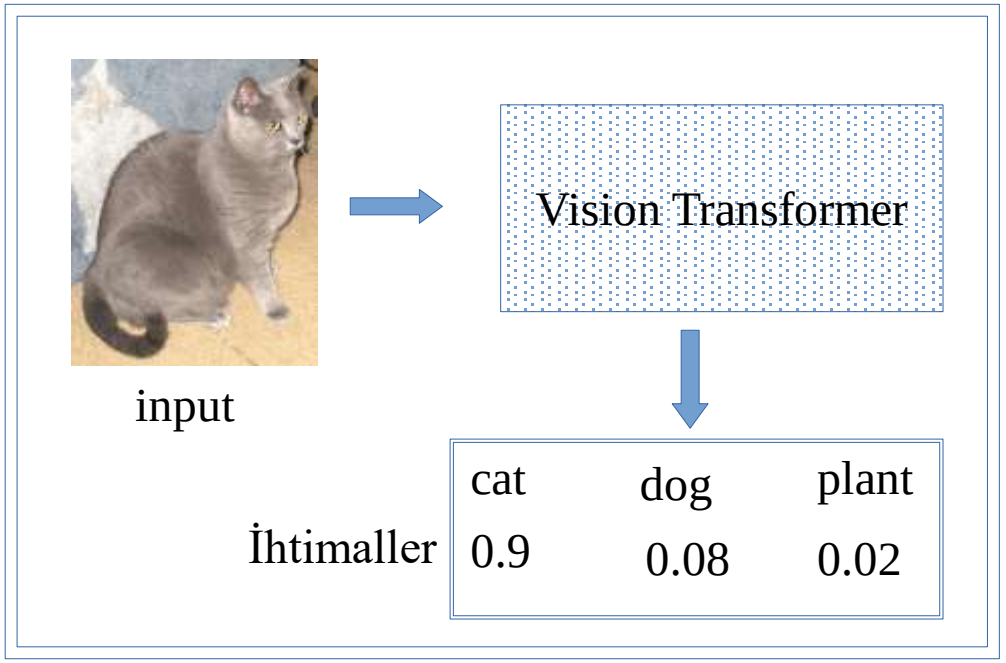
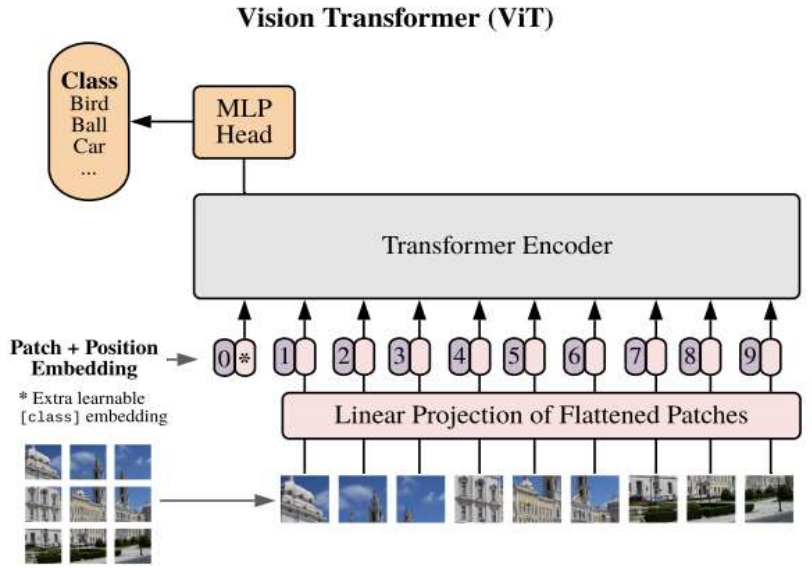
VISION TRANSFORMERS

Nedir?

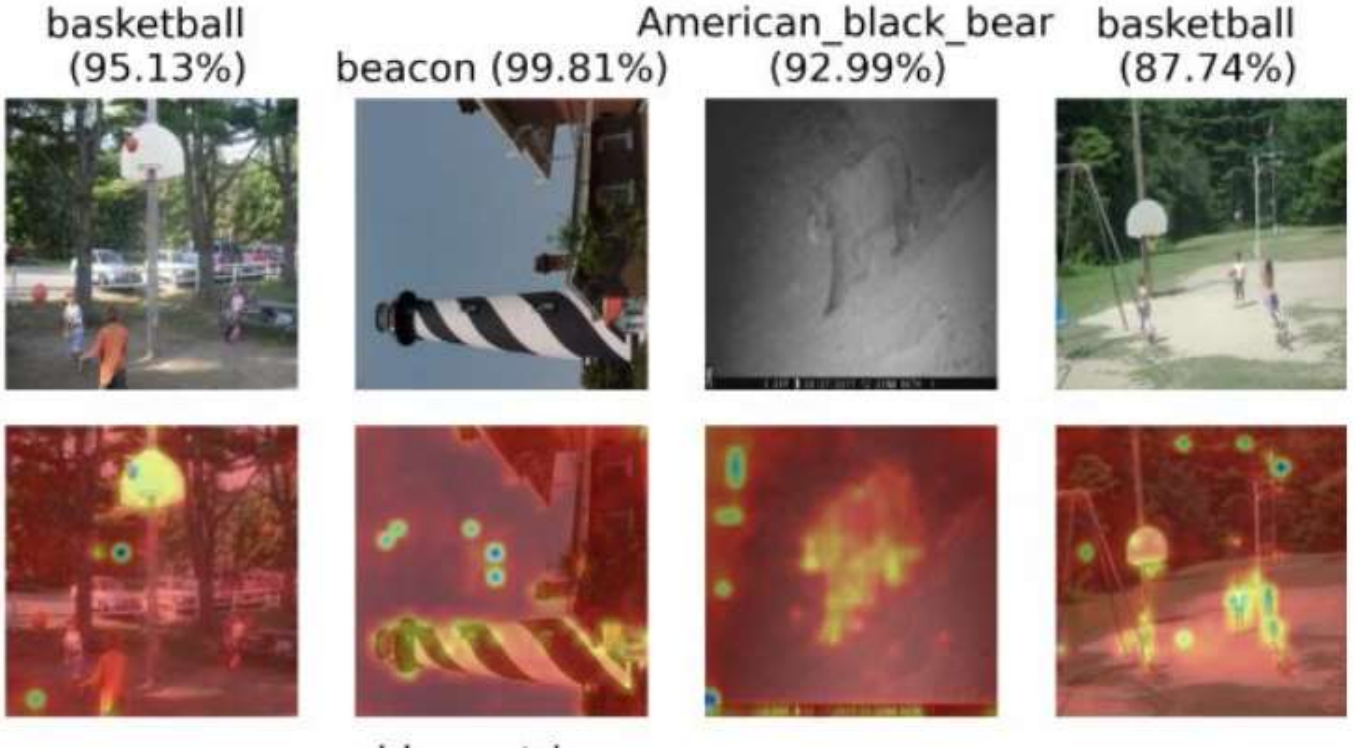
Bir görüntüyü daha küçük görüntüler parçalara ayırarak bu küçük görüntülerin birbirleriyle nasıl ilişkili olduğunu öğrenir ve bu ilişkilendirme tekniği ile görüntüde neler olduğunu belirleme, nesneleri konumlandırma ve hatta görüntünün farklı segmentlerini tanımlama gibi çeşitli görevleri gerçekleştirmek için kullanır. Bu yaklaşım, görüntü parçalarının kendi çapında özelliklerini ve bu parçaların bir araya gelerek tüm görüntüyü nasıl oluşturduklarının bütünsel bağlamını yakalamalarını sağlar.





Anlamdırmayı yaparken görüntü parçaları bilgisayarın öğrenebileceği formata çevirilir. Bu görüntü parçalarının konumları ve anlamların saklanacağı embedding denen vector, modele verilir.

Model bu görüntü parçalarını konumlarına ve anlam vektörlerine göre ilişkilendirir. Eğitim sırasında parametreleri değiştirir. Ve son olarak model çıktı olarak görüntünün hangi sınıflara ait olabileceğini gösteren ihtimalleri çıktı olarak verir.



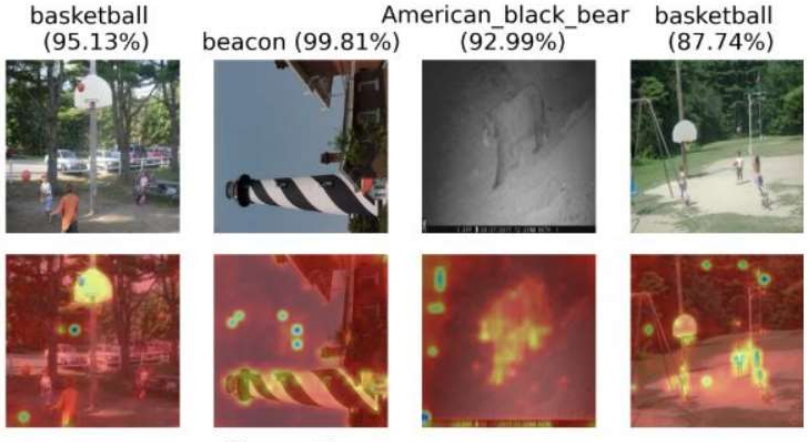
VISION TRANSFORMERS

Vision Transformers çeşitli görüntü anlama ile ilgili görevleri yerine getirebilir, bunlar arasında:

Görüntü Sınıflandırma: Bir görüntünün hangi kategoriye ait olduğunu belirleme (örneğin, bir fotoğrafın köpek veya kedi olup olmadığını tanımlama).

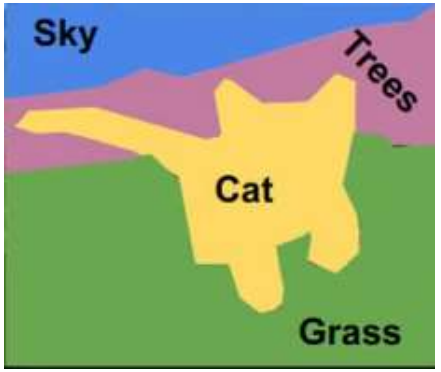
Nesne Tespiti: Bir görüntü içindeki nesneleri ve sınırlarını belirleme.

Segmentasyon: Farklı bileşenleri anlamak için bir görüntüyü segmentlere ayırma (örneğin, kalabalık bir sahnede her bireyi ayırt etme).



Classification

Object Detection



Segmentation

Bazı modifikasyonlarla video anlama ve sınıflandırma dahi yapabilir.

Anomaly Detection:

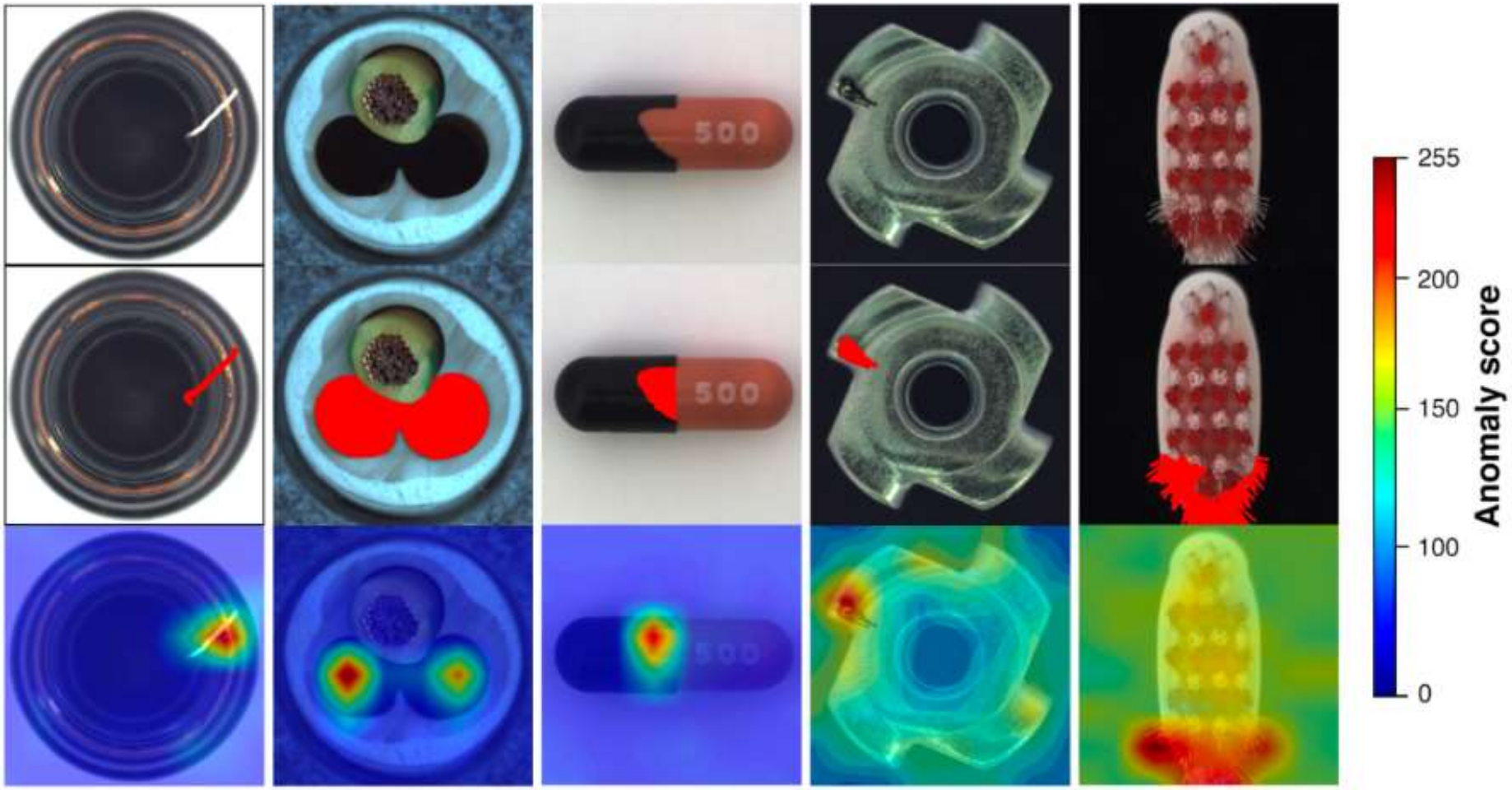


Image Captioning

A person on a dirt bike jumping in the air.

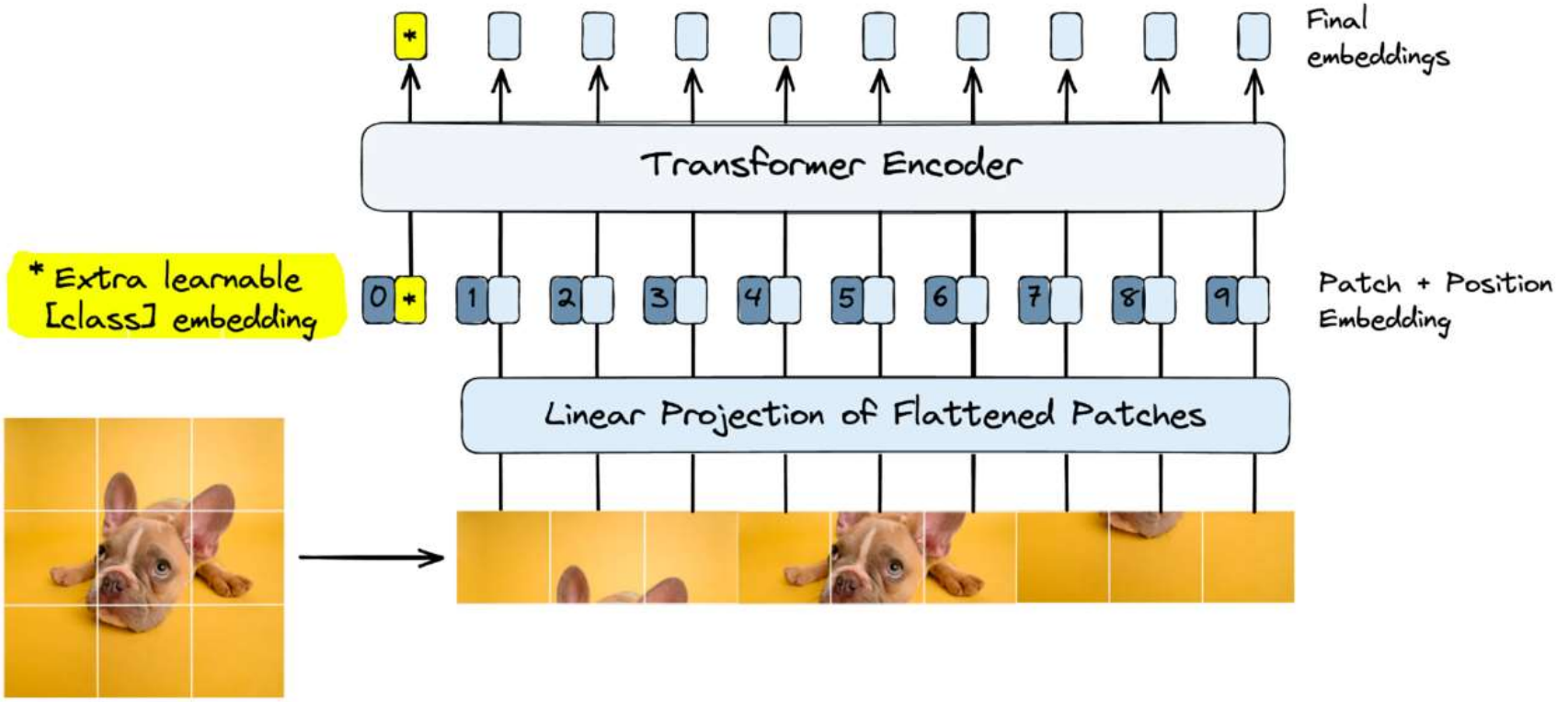
A cat sitting on the edge of a sink.

A person on a motorcycle is racing on a track.

Image Caption Generation using a ViT

Aşamalar

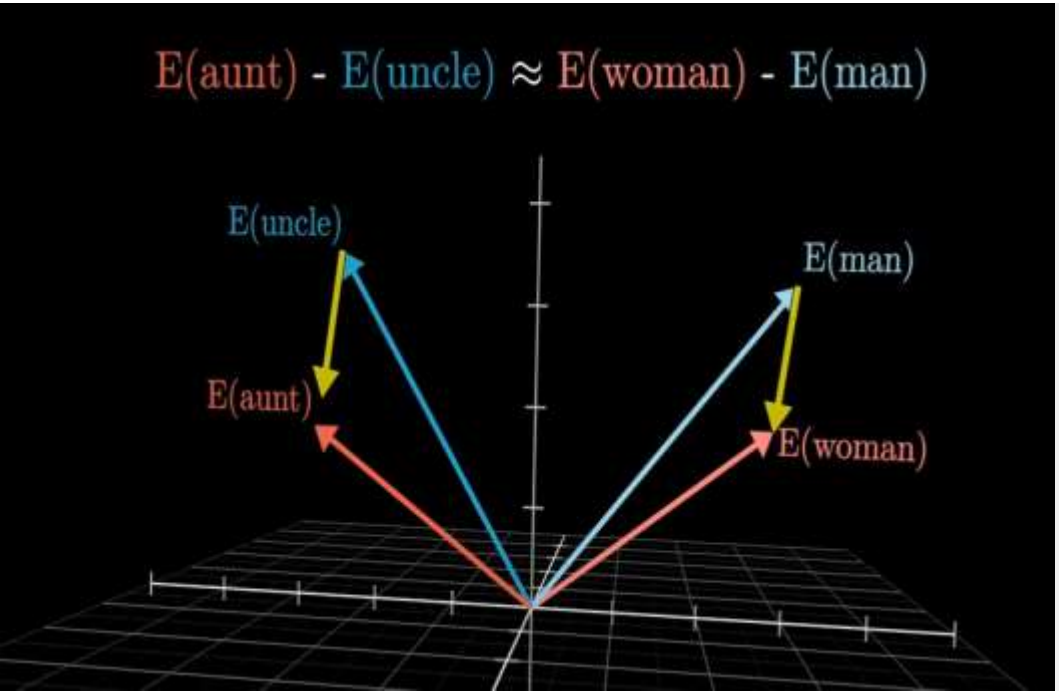
- Resmi Sabit Boyutlarda Parçalara Bölme:
- Görüntü Parçalarının Düzleştirilmesi ve Linear Projection:
- Positional Embedding eklenmesi:
- Class token eklenmesi:
- Transformera inputların verilmesi:
- Transformerın outputu ve sınıflandırma:



ViT process with the learnable class embedding highlight (left).

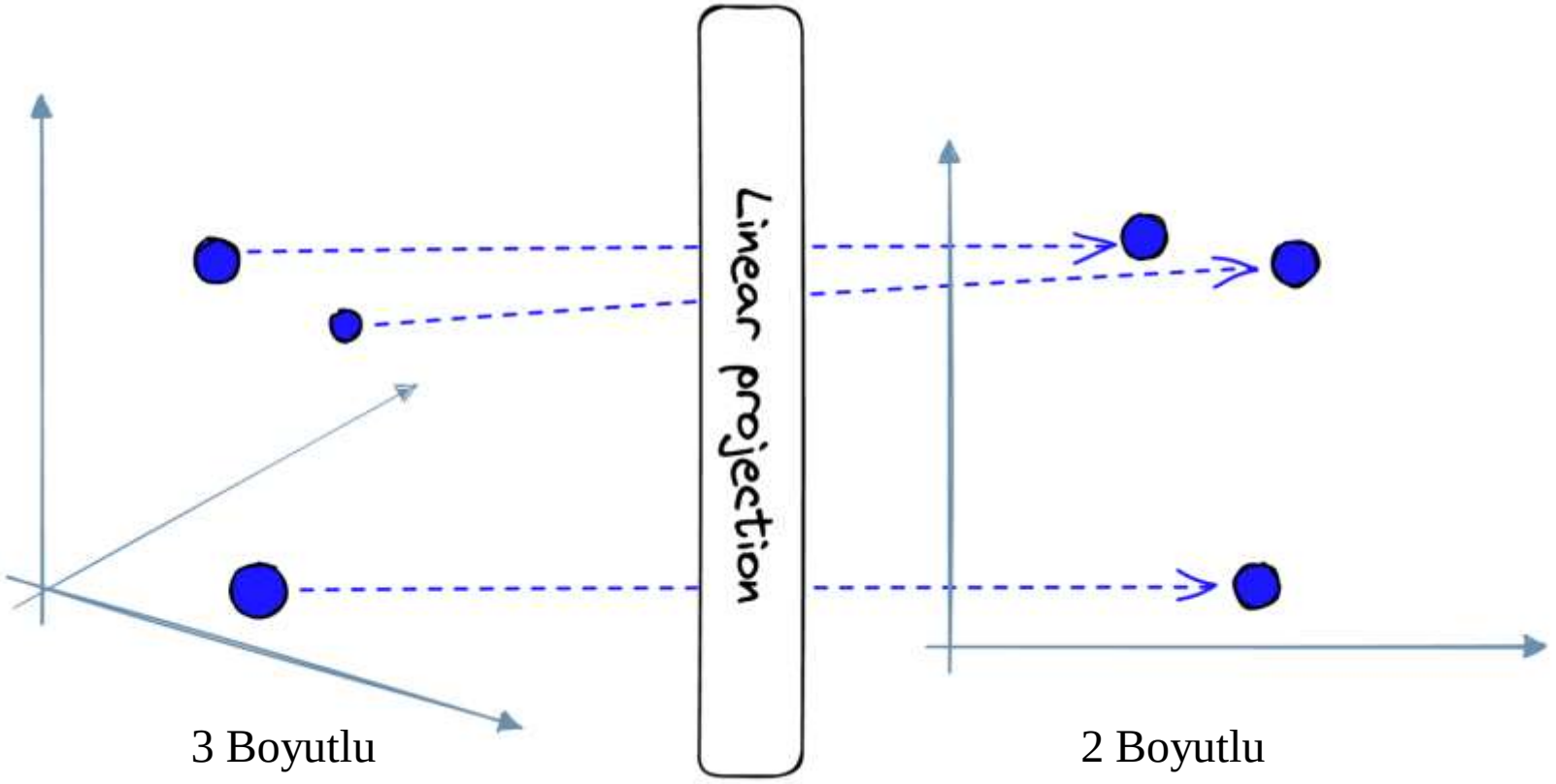
Embedding Nedir?

- Kelimenin yada bizim durumumuzda resim parçalarının çok boyutlu vektörlerle ifade edilmesidir.
- Anlamın vektör ile ifade edilmesi denebilir.
- Embeddingler eğitim başında rastgele sayılardır, eğitim sırasında öğrenilir.



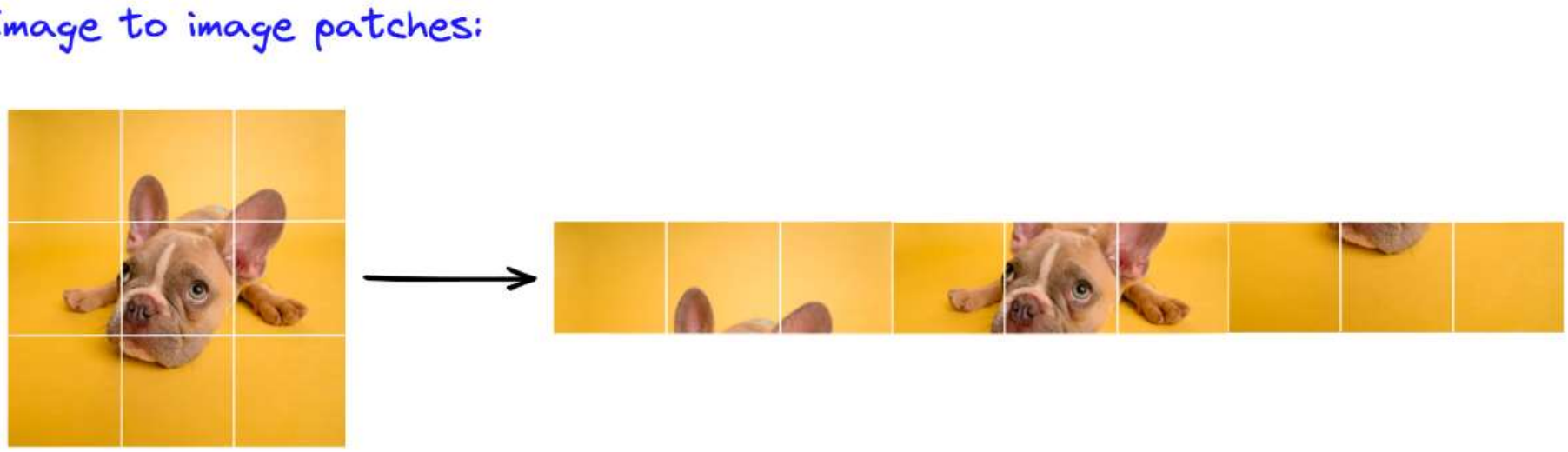
Linear Projection Nedir?

- Linear projection, verinin boyutunu değiştirme işlemidir.
- Bu işlem, Vision Transformers modelinde resim parçalarının daha az karmaşık bir temsilini oluşturmak için kullanılır.
- Daha az boyutlu veri kullanmak modelin verimli çalışmasını sağlar



Resimlerin Patch'lere Ayrılması

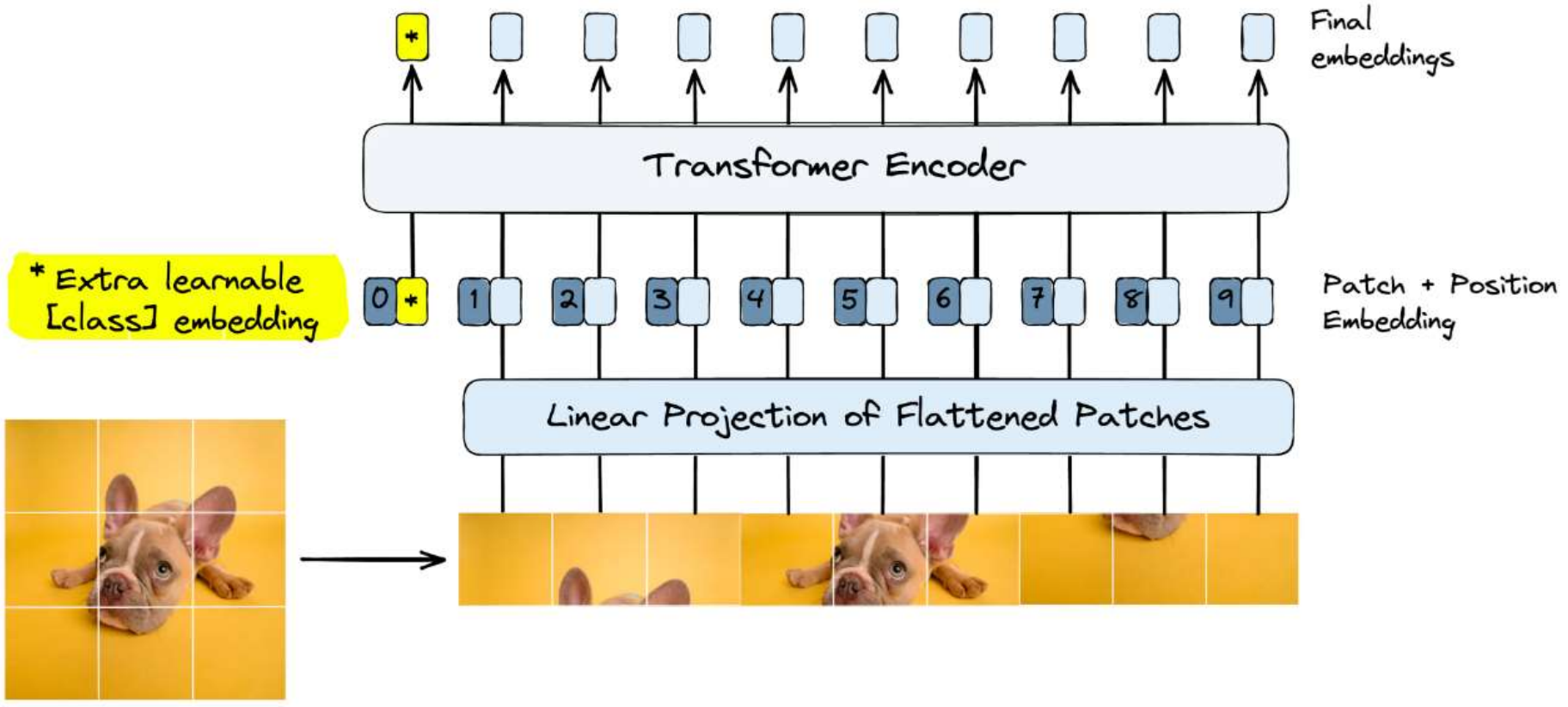
ViT, bir görüntüyü sabit boyutlu parçalara (patch) ayırarak başlar. Örneğin, bir görüntüyü 9 eşit parçaya bölebiliriz. Bu yaklaşım, her bir pikselin diğer tüm piksellerle karşılaştırılmasını gerektiren ve hesaplama açısından oldukça maliyetli olan quadratic işlemlerin daha az yapılmasını sağlar. Eğer her pikselin embedding'ini almaya çalışırsak, bu çok fazla parametre ve hesaplama gerektirir. Bunun yerine her bir patch'in embedding'ini hesaplatmak daha verimli.



NLP transformers and ViT both split larger sequences (sentences or images) into tokens or patches.

Patch'lerin Flattening ve Linear Transformation İşlemi

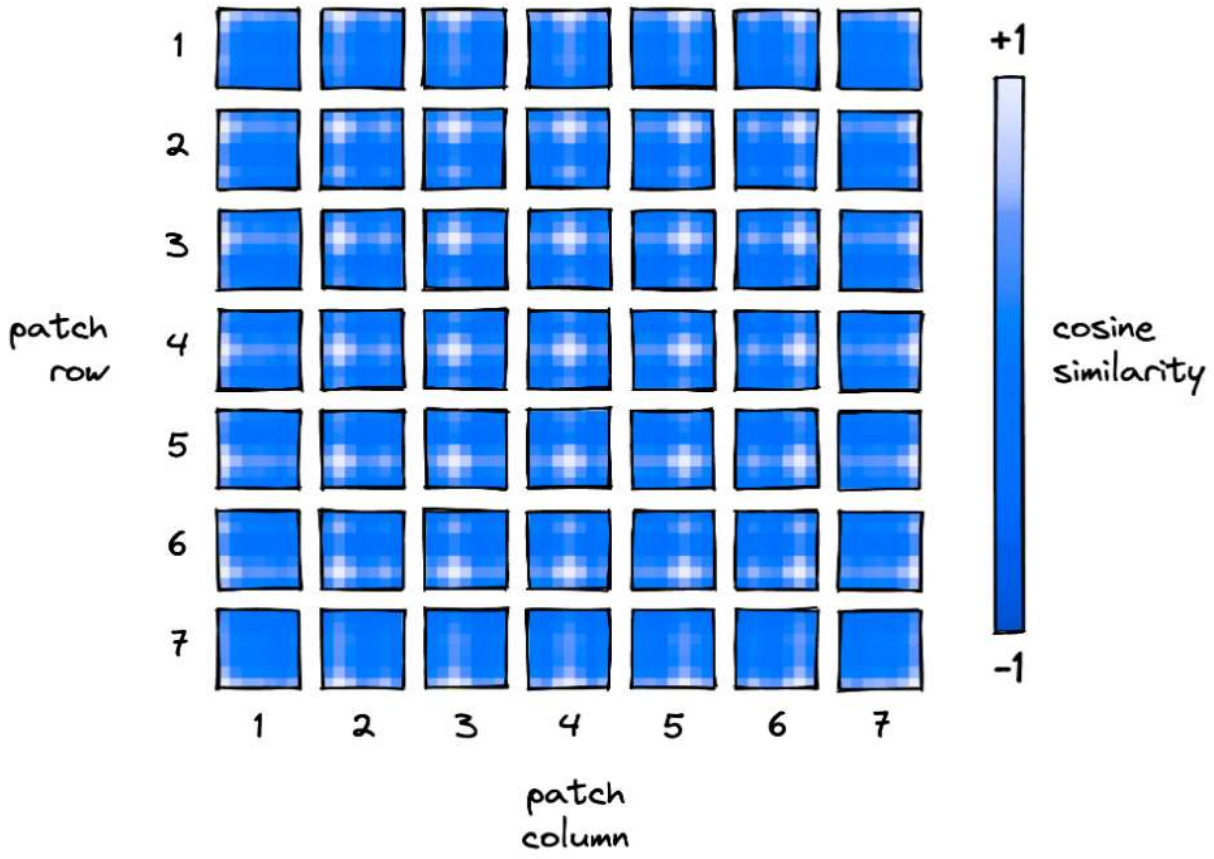
- Her bir patch düzleştirildikten sonra, elimizde çok boyutlu vectorler kalıyor. Bu vektörler daha sonra bir doğrusal projeksiyon aracılığıyla daha düşük boyutlu bir embedding vektöre dönüştürülür. Bu işlem, modelin eğitimi sırasında daha az parametre kullanmasını sağlar ve hesaplama verimliliğini artırır.



ViT process with the learnable class embedding highlight (left).

Positional Embeddings

- Konum ve sıra bilgisinden yararlanmak için üretilen parametrelerdir.
- Eğitim sırasında positional embeddingler, komşu positional embeddingler ile yüksek benzerlik gösterdikleri benzerlikleri aşağıda göründüğü üzere yüksek çıkar.



Cosine similarity between trained positional embeddings. Adapted from [1].

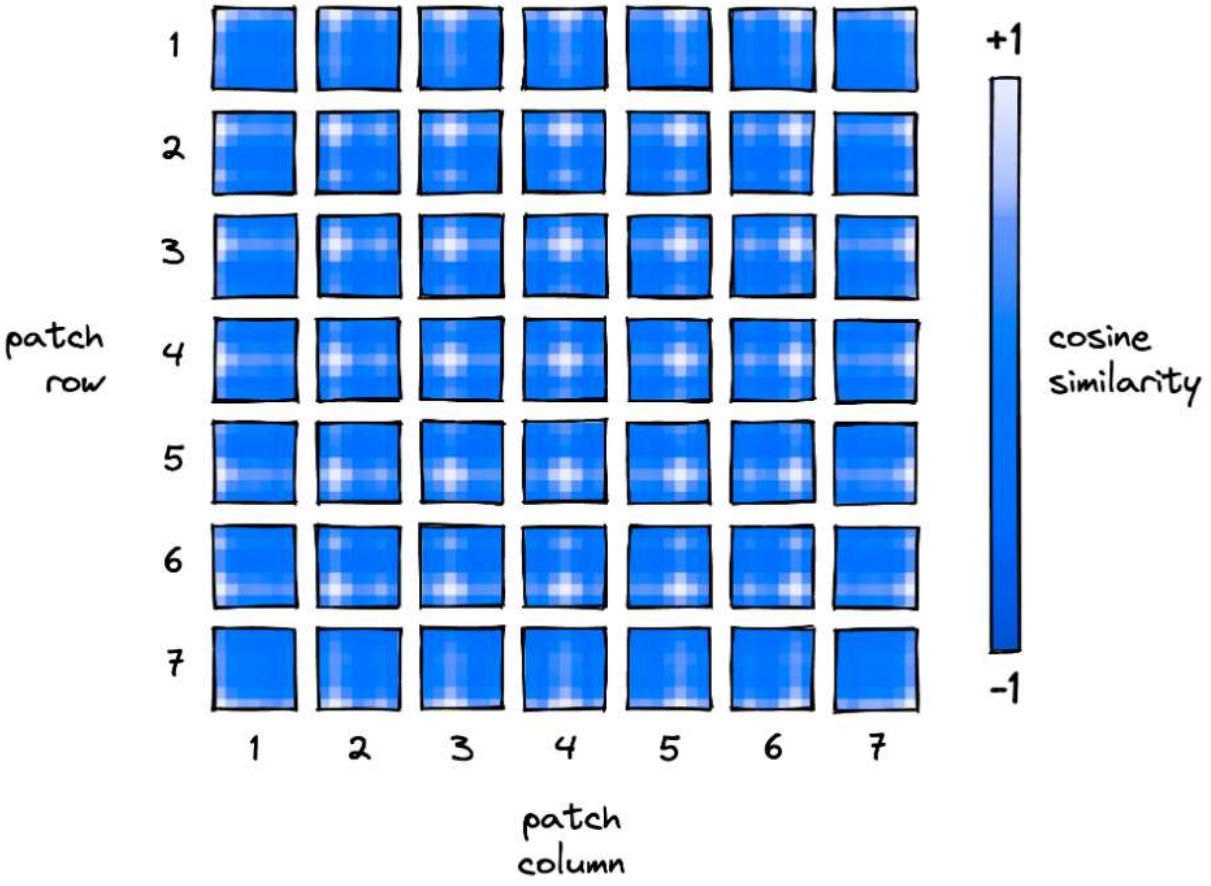
Positional Embedings neden gereklidir?

- Patchler karmaşık sıralanırsa orijinal görüntünün anlamı kaybolur.

- Bu yüzden görüntü parçalarının sıra ve konumlarını belirten positional embeddingler kullanılır.

Positional Embeddings Eklenmesi:

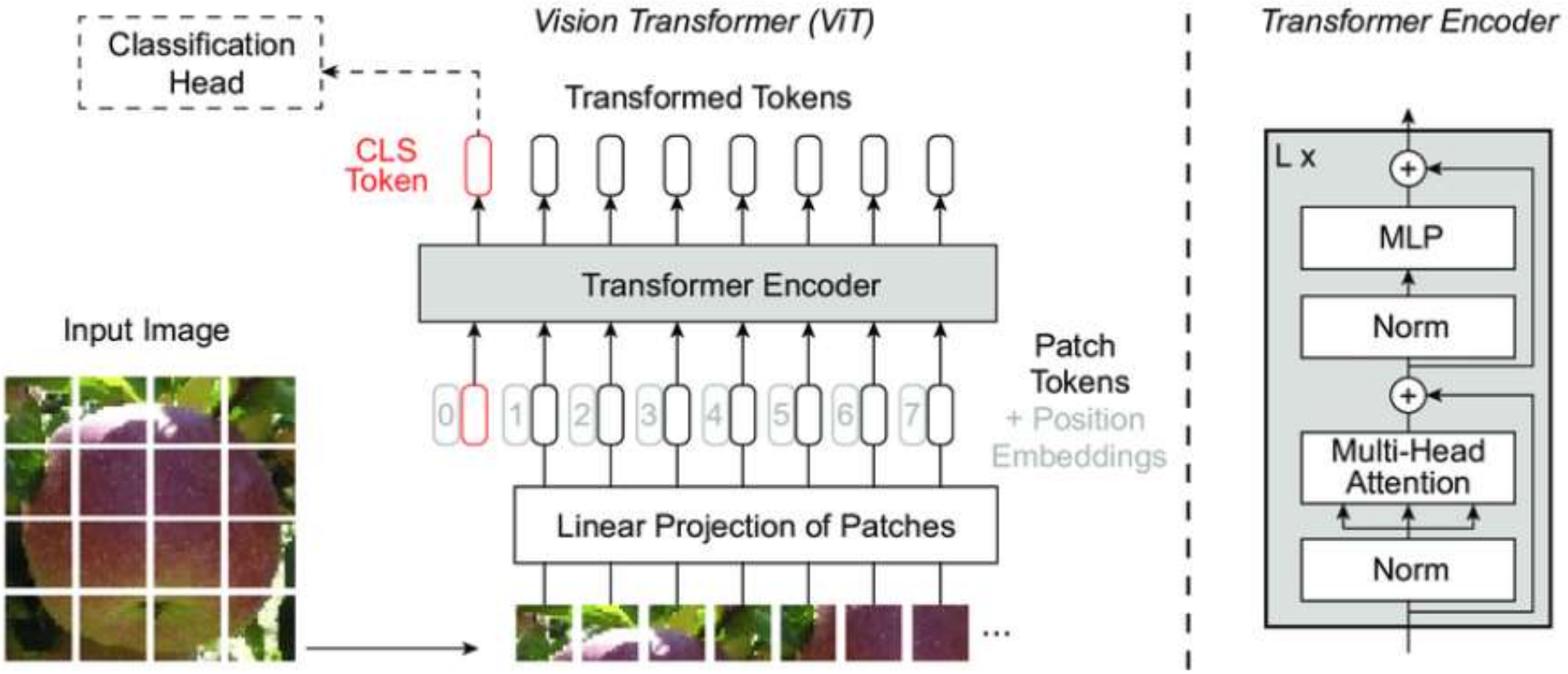
- Her patch'e bir positional embedding eklenir.Bu positional embeddingler, patch embeddingler ile aynı boyuta sahip vektörlerdir. Bu embeddingler başlangıçta rastgele sayılarla başlar (çoğu zaman) ve eğitim sürecinde optimize edilir.
- Bu Positional Embeddingler, Patch embeddingler ile toplanır ve daha sonra transformera yollanır.



Cosine similarity between trained positional embeddings. Adapted from [1].

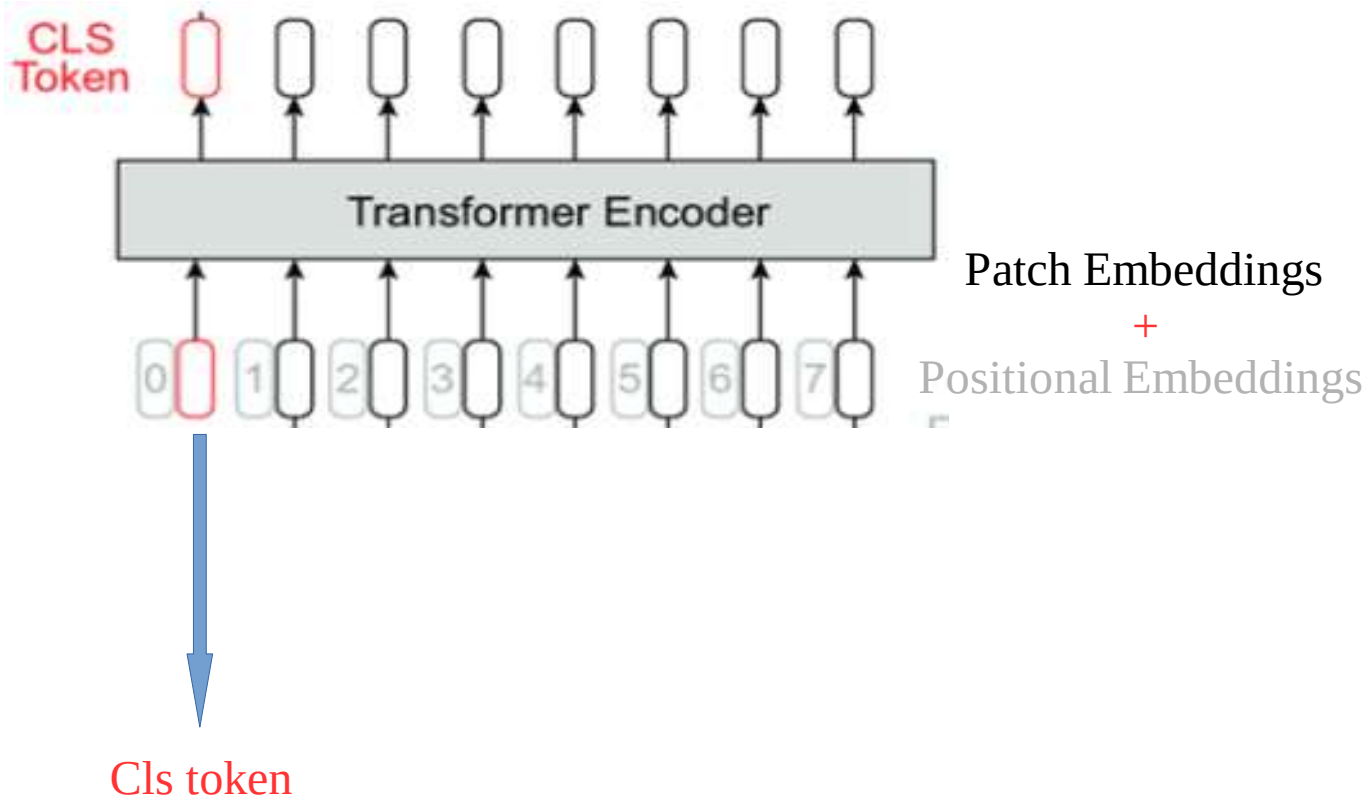
Class Token Nedir?

- Transformera rastgele parametrelerle başlatılan bir class token yollanır. Bu token eğitim sırasında değişir ve girdinin hangi sınıfa ait olduğunu tahmin etmek için kullanılır.
- Classification head girdi olarak class token alır ve output olarak sınıflandırma sonucunu verir.



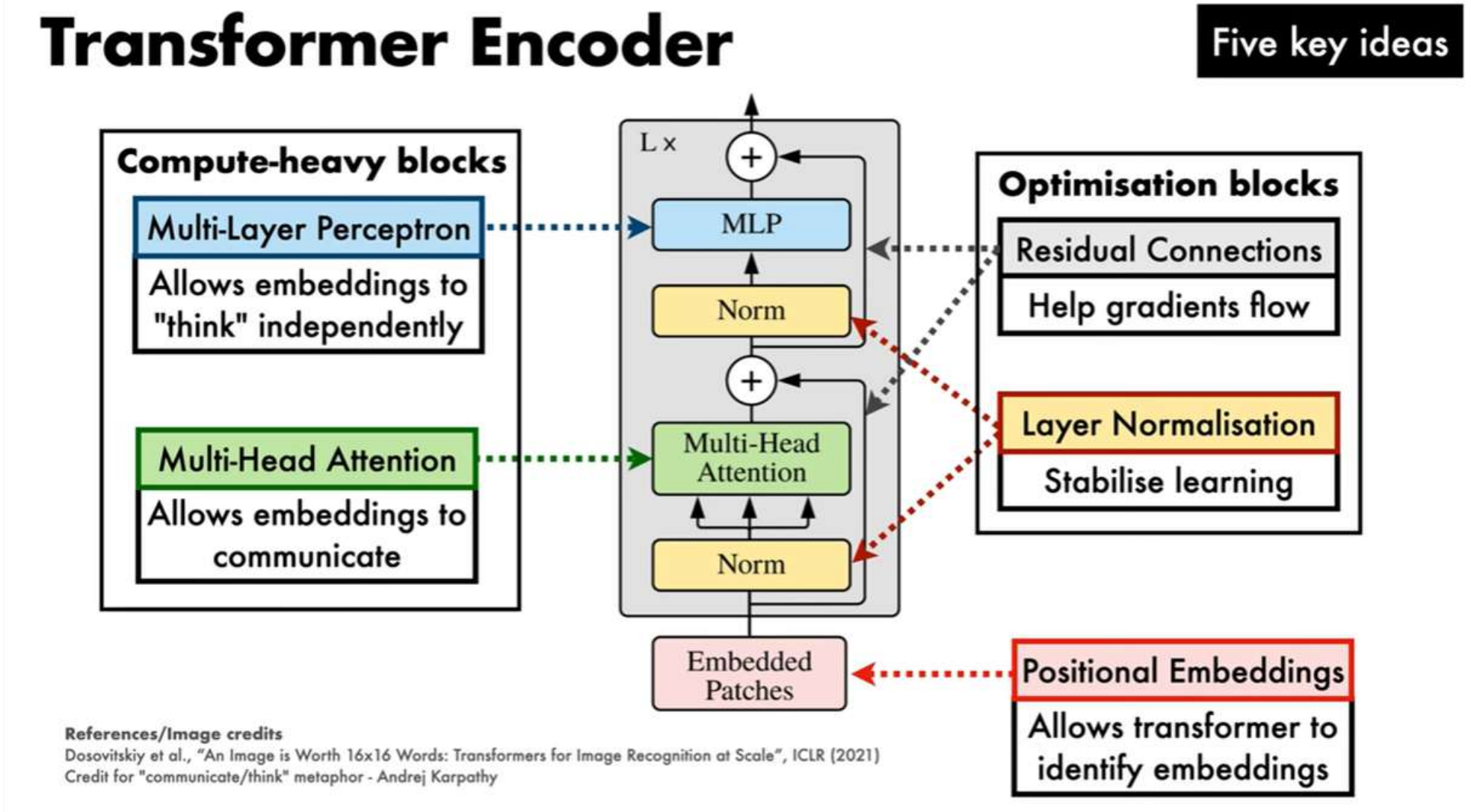
Transformer İnputları Nelerdir?

- Positional embeddings, patch embeddings ve class token encoder'a girdi olarak verilir.



Peki Transformer ne yapar?

- İnput olarak aldığı patch embedding ve positional embedding kullanarak patchlerin birbiri ile ilişkilendirilmesini sağlar.
- Öğrenilebilir parametreleri eğitim sırasında değiştirir



Attention Nedir?

Modelin sınıflandırma yapması için en önemli olduğunu düşündüğü pixellerdir.

Patchlerin birbiri ile ilişkilendirilmesi ile elde edilir.

Model görüntüdeki cismi anlamaya çalışırken attentionın yüksek olduğu pixellere daha çok önem verilir.

Shown: self-attention of the CLS token on the last layer

Attention Nasıl Hesaplanır?

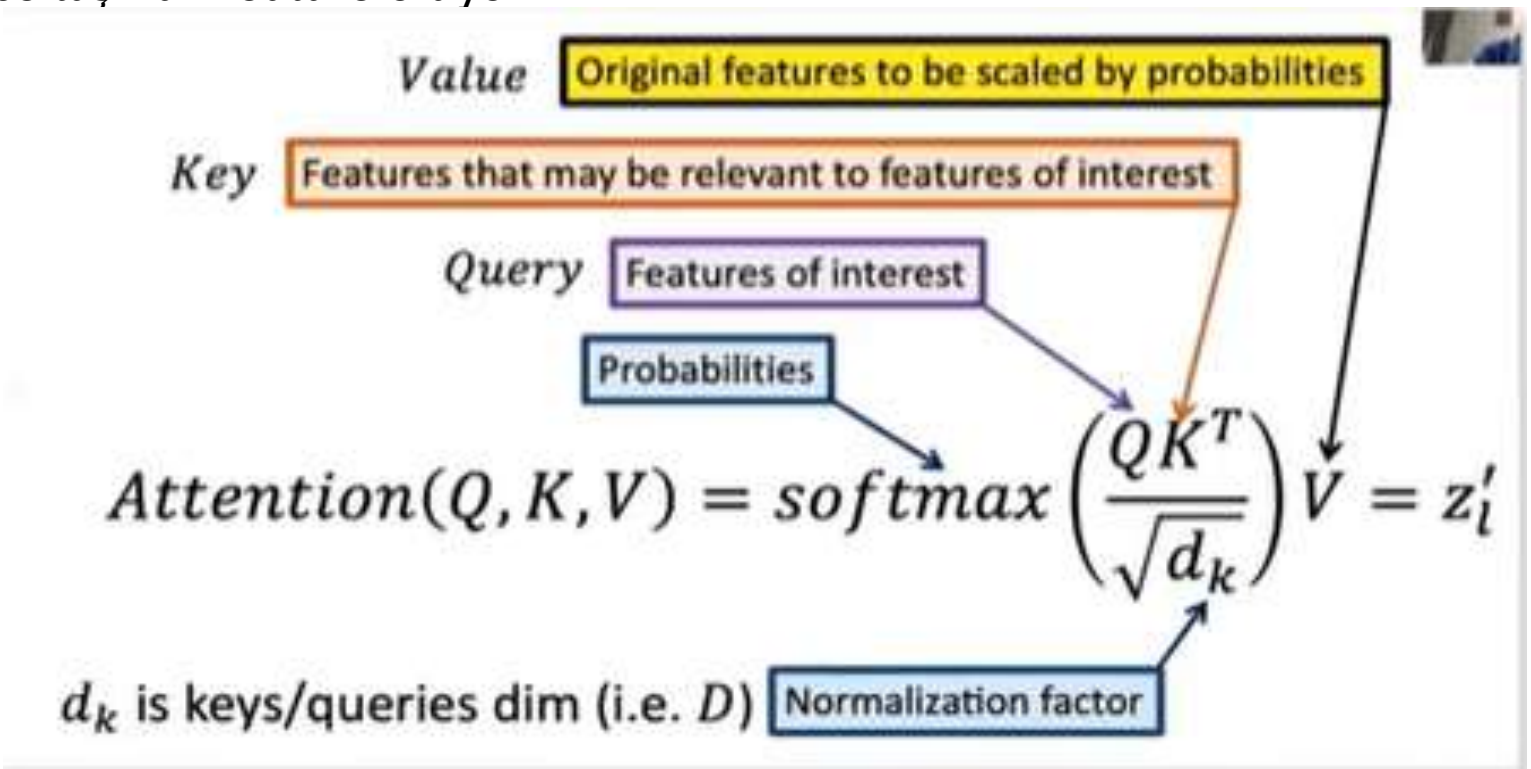
Query, Key ve Value rastgele başlatılan embeddinglerdir.

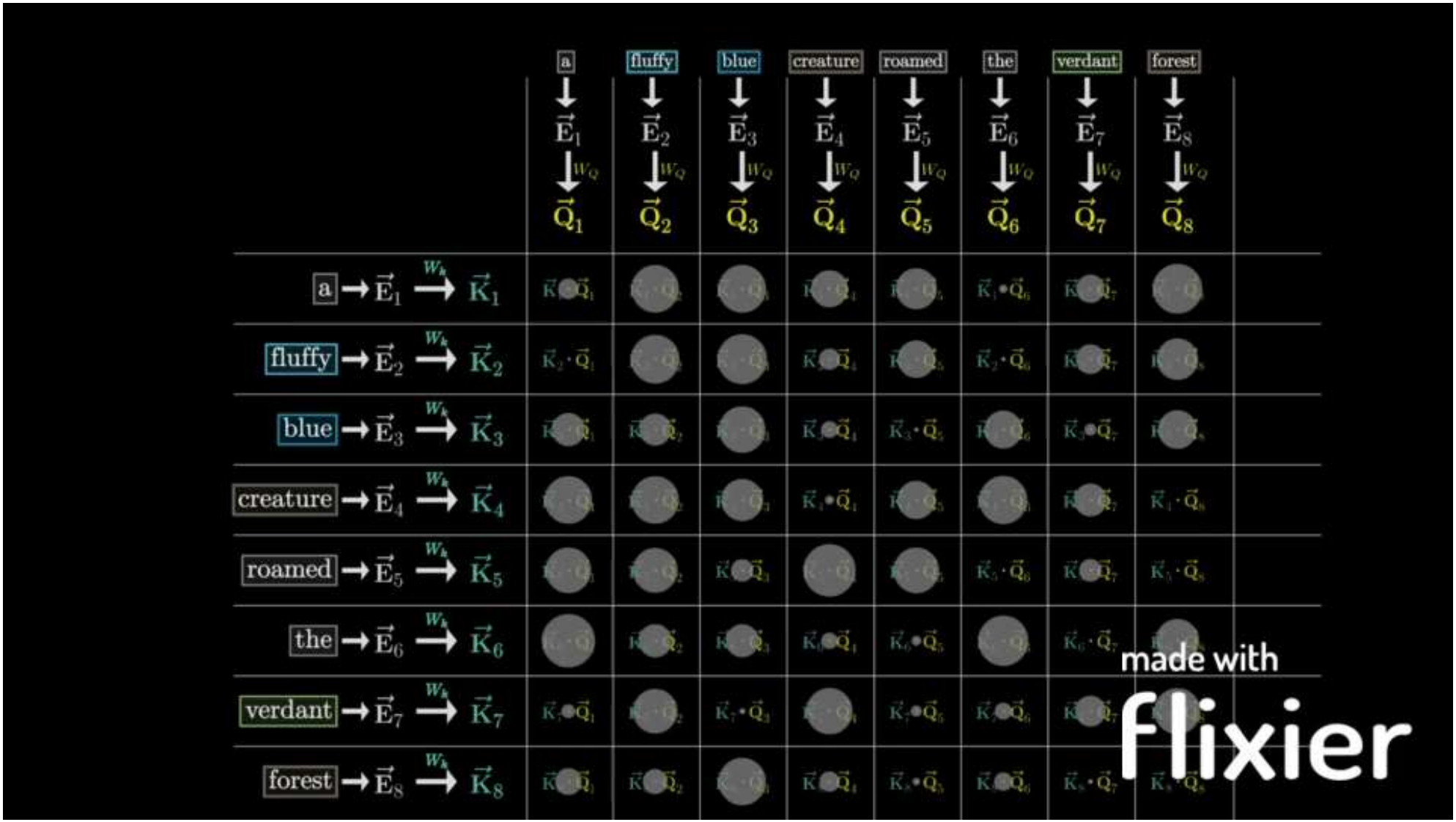
Model eğitim sırasında önemli bilgi taşıyan ilişkileri bu parametrelerle oynayarak bulmaya çalışır.

Query feature arıyor
Key feature'ını belirtiyor.

Bunlar benzer vektörler ise çarpımlarıda büyük oluyor.

Value ise taşınan feature oluyor.




a
fluffy
blue
creature
roamed
the
verdant
forest
made with
flixier

Çok büyük sayıları önlemek için Scaling(ölçeklendirme) yapılır. Boyut sayısının kökü ile bölünür.
Softmax ise her sütündaki değerlerin toplamının 1 e eşit olmasını sağlar.

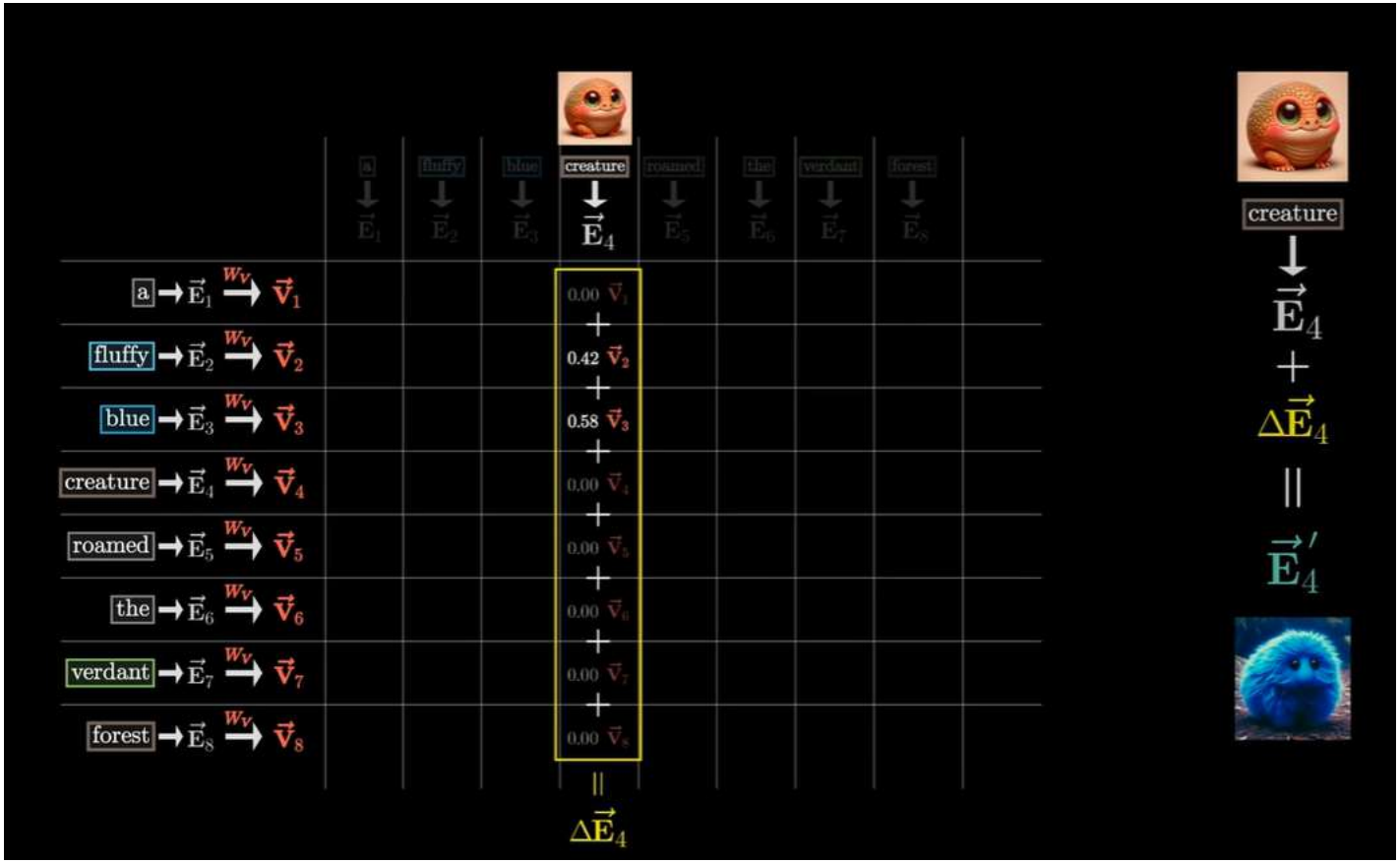
$$\text{Attention}(Q, K, V) = \text{softmax}(\boxed{\frac{QK^T}{\sqrt{d_k}}})V$$

	Q_1	Q_2	Q_3	Q_4	Q_5	$\cdots$	Q_n
K_1	$\frac{Q_1 \cdot K_1}{\sqrt{d_k}}$	$\frac{Q_2 \cdot K_1}{\sqrt{d_k}}$	$\frac{Q_3 \cdot K_1}{\sqrt{d_k}}$	$\frac{Q_4 \cdot K_1}{\sqrt{d_k}}$	$\frac{Q_5 \cdot K_1}{\sqrt{d_k}}$	$\cdots$	$\frac{Q_n \cdot K_1}{\sqrt{d_k}}$
K_2	$\frac{Q_1 \cdot K_2}{\sqrt{d_k}}$	$\frac{Q_2 \cdot K_2}{\sqrt{d_k}}$	$\frac{Q_3 \cdot K_2}{\sqrt{d_k}}$	$\frac{Q_4 \cdot K_2}{\sqrt{d_k}}$	$\frac{Q_5 \cdot K_2}{\sqrt{d_k}}$	$\cdots$	$\frac{Q_n \cdot K_2}{\sqrt{d_k}}$
K_3	$\frac{Q_1 \cdot K_3}{\sqrt{d_k}}$	$\frac{Q_2 \cdot K_3}{\sqrt{d_k}}$	$\frac{Q_3 \cdot K_3}{\sqrt{d_k}}$	$\frac{Q_4 \cdot K_3}{\sqrt{d_k}}$	$\frac{Q_5 \cdot K_3}{\sqrt{d_k}}$	$\cdots$	$\frac{Q_n \cdot K_3}{\sqrt{d_k}}$
K_4	$\frac{Q_1 \cdot K_4}{\sqrt{d_k}}$	$\frac{Q_2 \cdot K_4}{\sqrt{d_k}}$	$\frac{Q_3 \cdot K_4}{\sqrt{d_k}}$	$\frac{Q_4 \cdot K_4}{\sqrt{d_k}}$	$\frac{Q_5 \cdot K_4}{\sqrt{d_k}}$	$\cdots$	$\frac{Q_n \cdot K_4}{\sqrt{d_k}}$
K_5	$\frac{Q_1 \cdot K_5}{\sqrt{d_k}}$	$\frac{Q_2 \cdot K_5}{\sqrt{d_k}}$	$\frac{Q_3 \cdot K_5}{\sqrt{d_k}}$	$\frac{Q_4 \cdot K_5}{\sqrt{d_k}}$	$\frac{Q_5 \cdot K_5}{\sqrt{d_k}}$	$\cdots$	$\frac{Q_n \cdot K_5}{\sqrt{d_k}}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

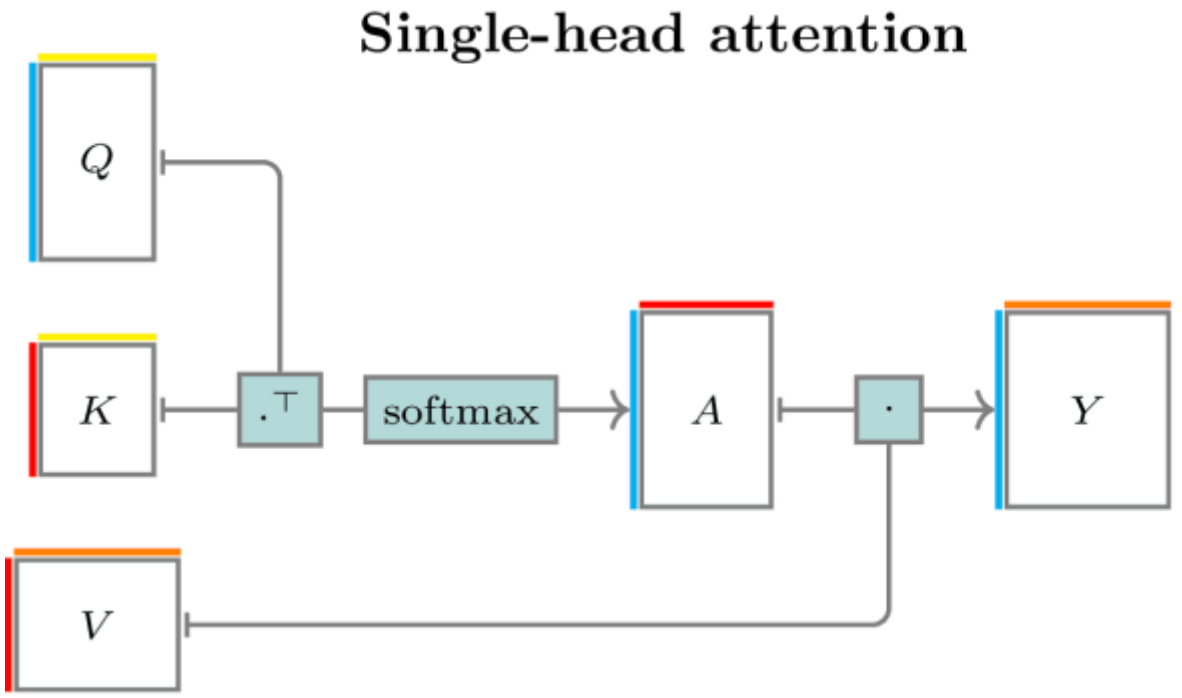
Bu kısma hangi patchlere daha çok önem verilmesi gerektiğini gösteren matrix diyebiliriz.

Bu kısmı value ile çarparak istediğimiz değişim vektörünü buluyoruz.



Single Head Attention

- Patch embeddinglerinin birbiri ile sadece bir kez iletişime geçme şansı var.
- Tek bir query, key ve value matrixi vardır.



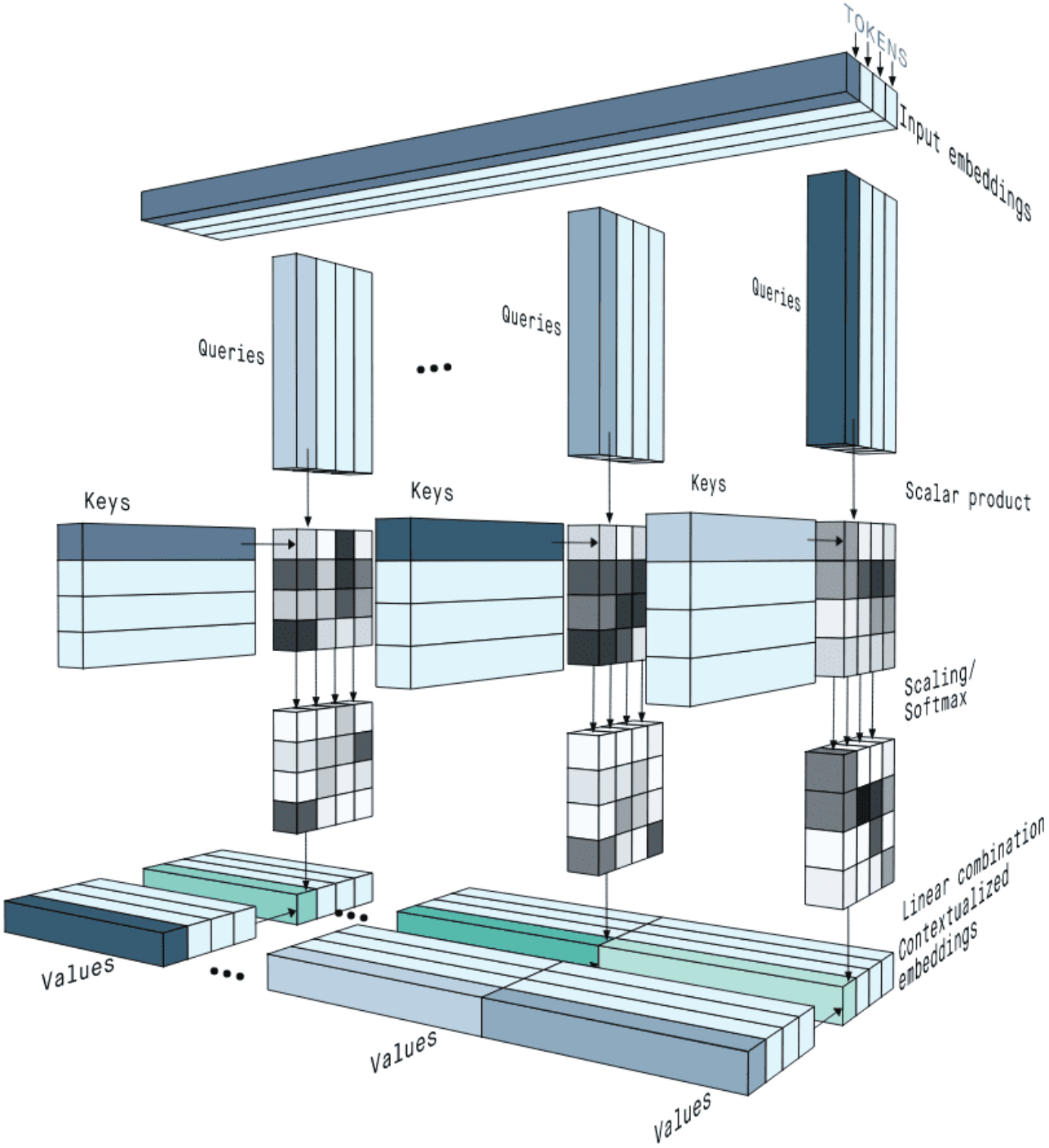
$$\text{Attention}(Q, K, V) = \text{softmax}_{\text{row}}\left(\frac{QK^\top}{\sqrt{d}}\right)V$$

Multi Head Attention

Bir patchin diğer patchlere birde fazla bilgi yollamasını sağlar.

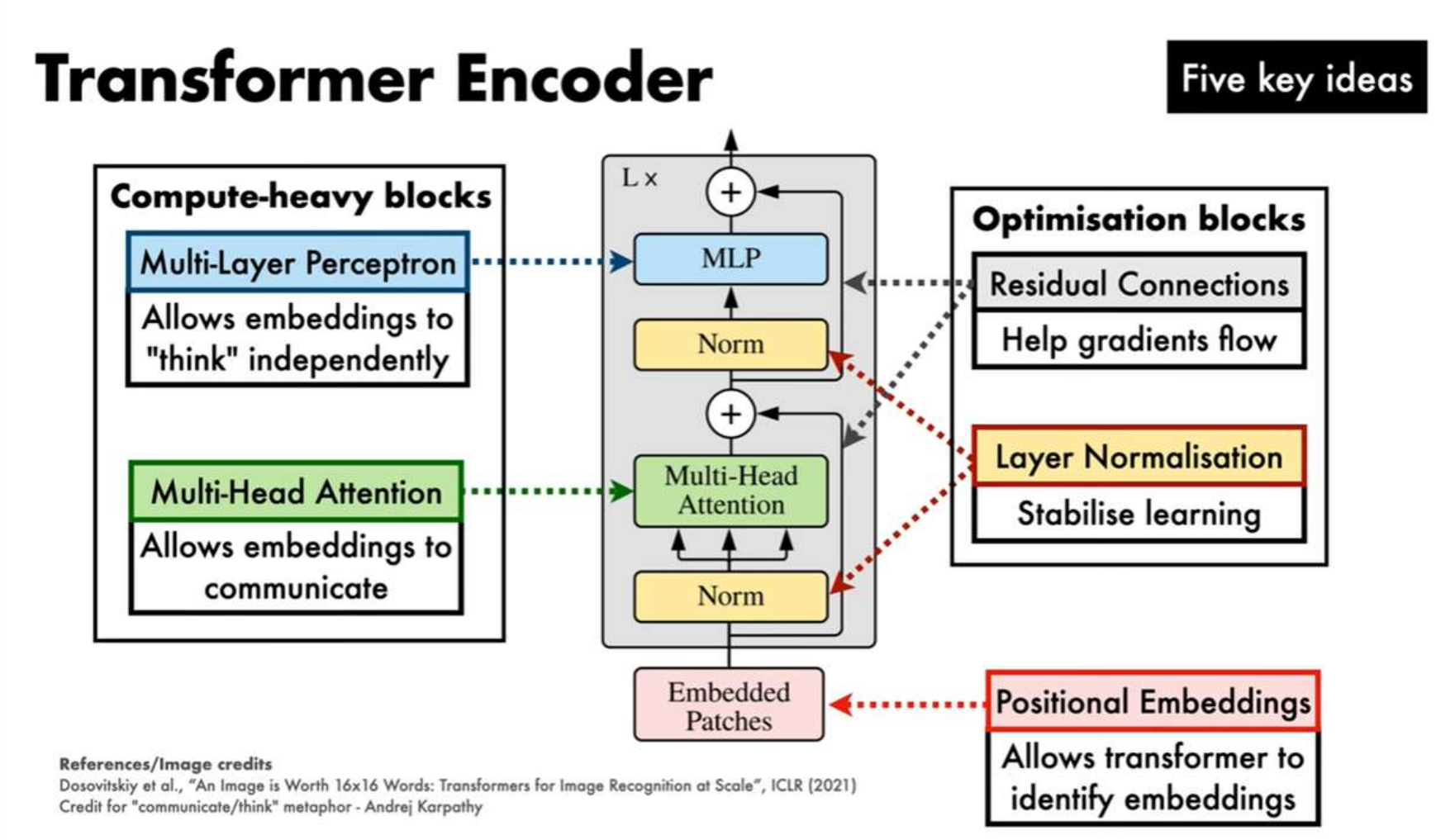
Her bir patch için attention head sayısı kadar query, key ve value matrixi oluşturulur.

Bu sayede model patchler arasındaki birden fazla ilişkiyi öğrenir.



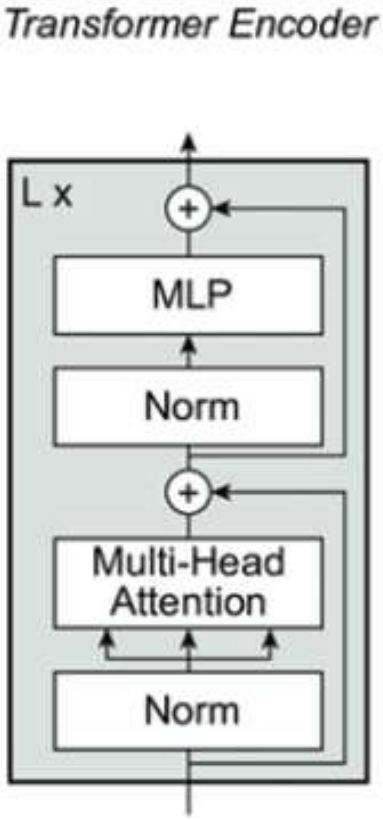
Residual Connections

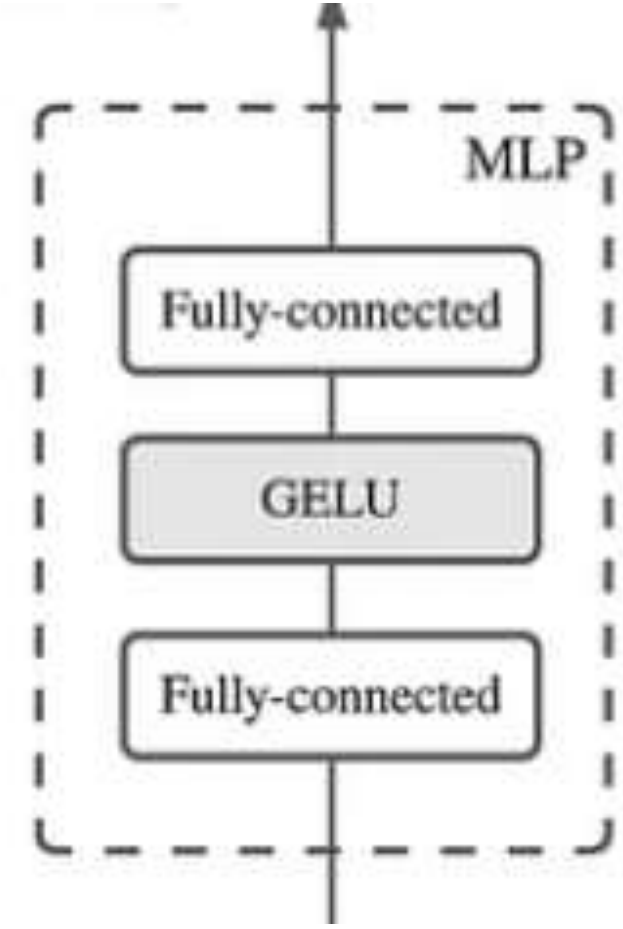
Stabilite , optimizasyon ve hız kazandırır.



MLP(Multi Layer Perceptron) Katmanı:

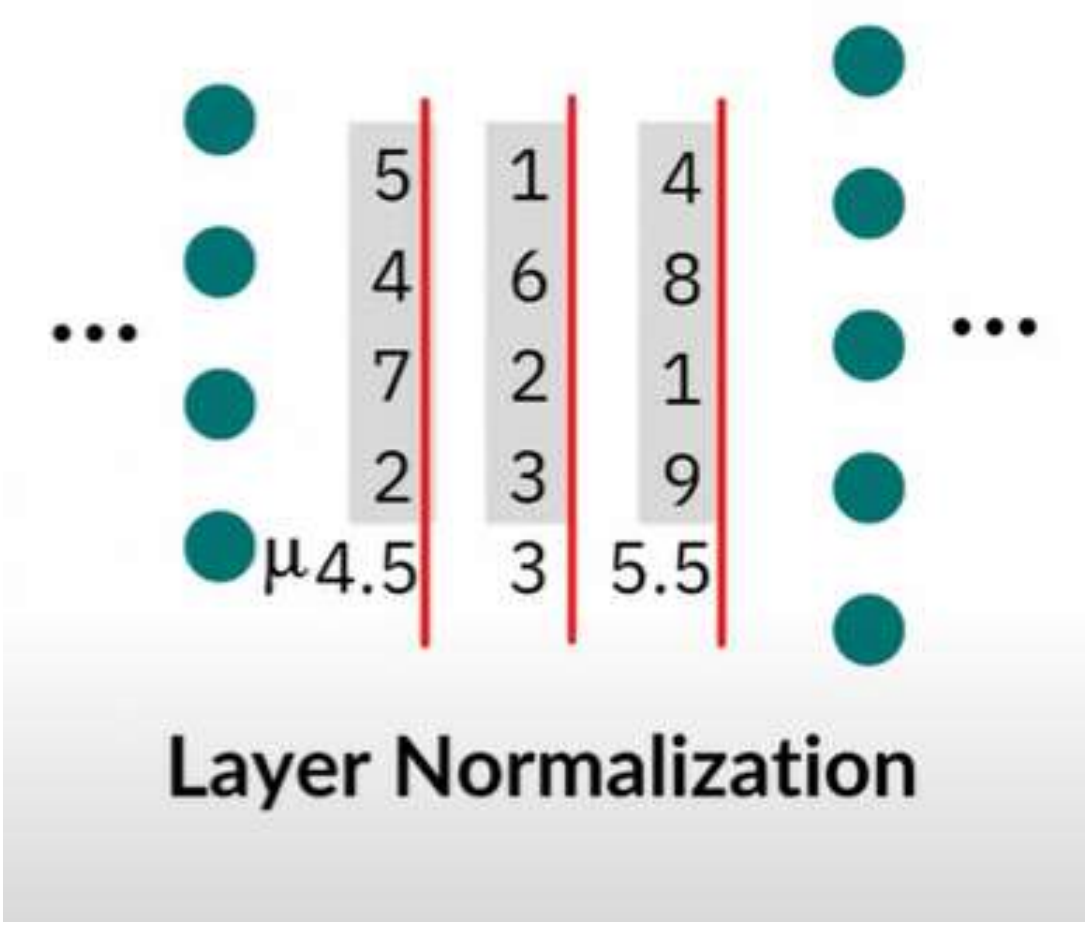
- Kompleks ve linear olmayan patternları yakalar.
- Attention blok'undan gelen outputu kullanır.





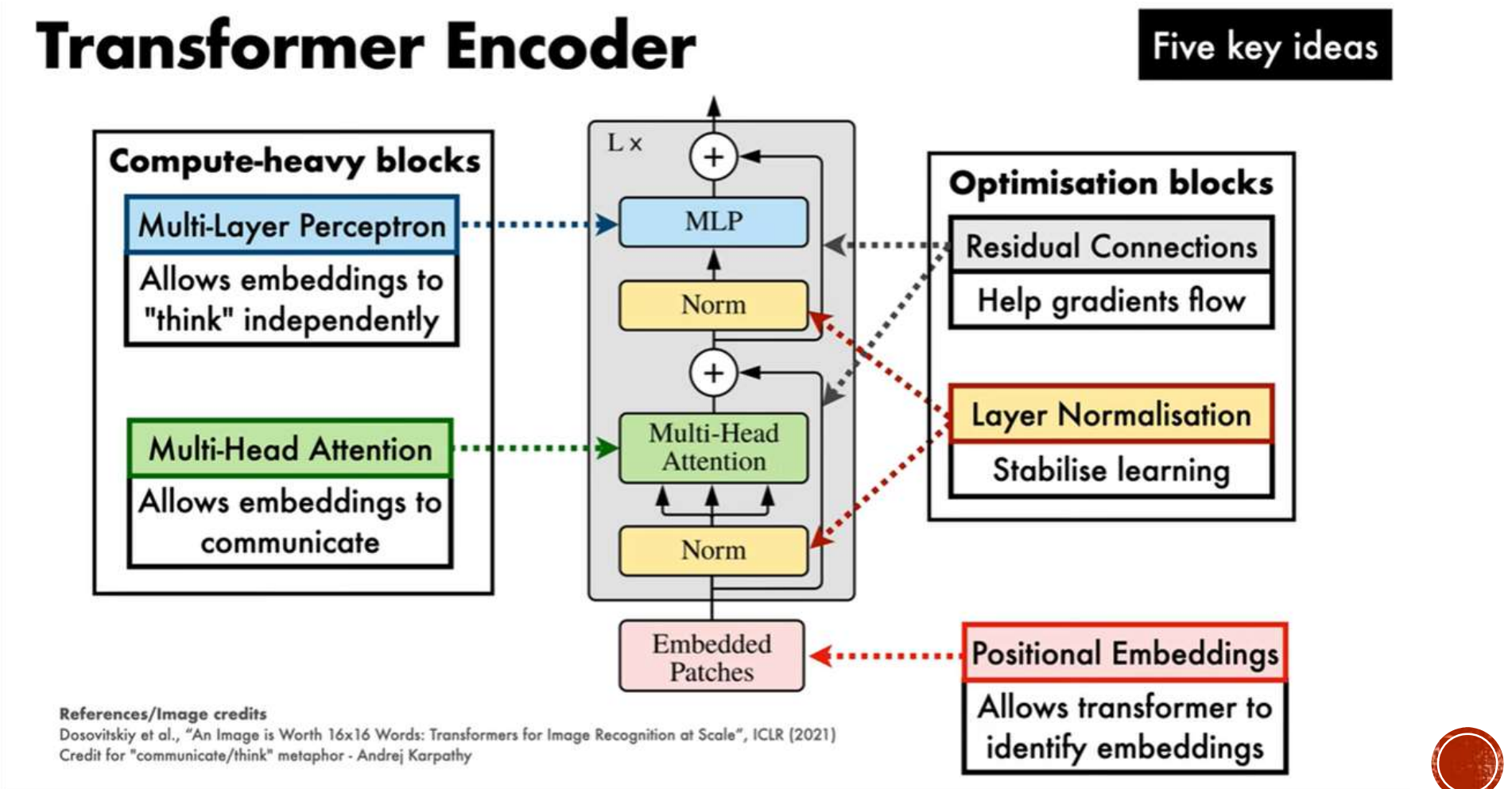
Layer Normalization

Layer normalization, her bir veri noktasını bağımsız olarak normalize ederek modelin performansını artırır ve stabil öğrenmesini sağlar.



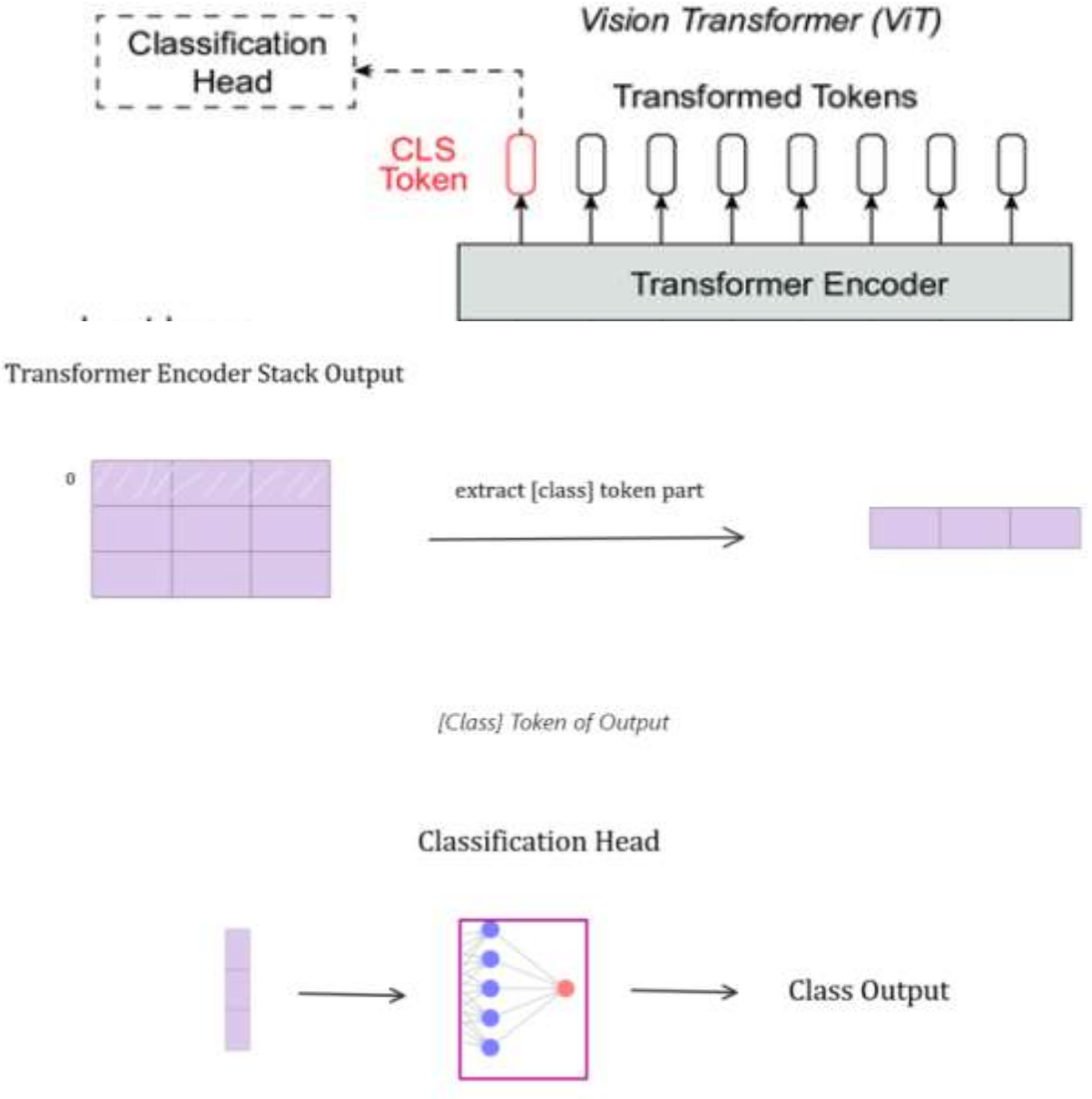
Skip connections:

- Skip connections (residual connections), eğitimin daha hızlı ve stabil olmasını sağlar.
- Vanishing gradient problemini çözer. Bu problem çok küçük sayıların birbiri ile çarpılmasıyla ortaya çıkar.

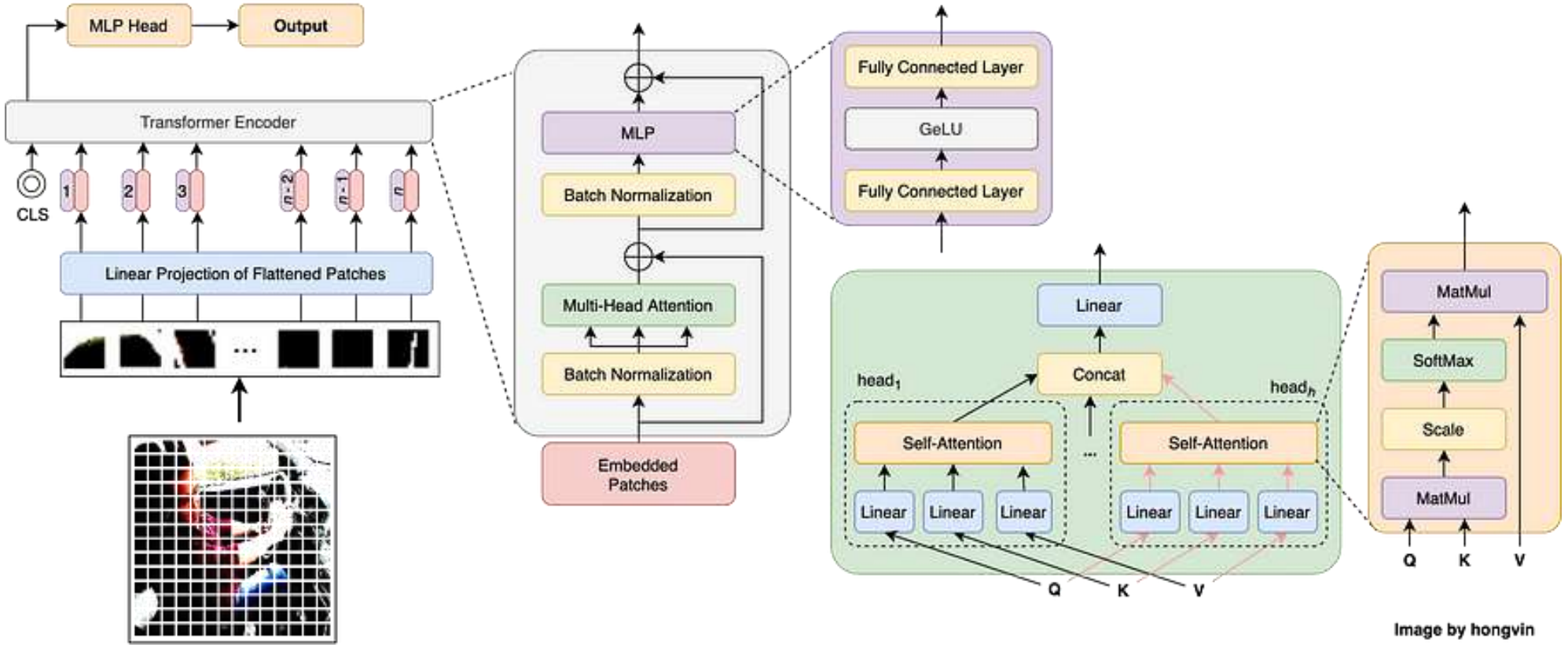


Classification Head:

Transformer Encoder outputlarından class token outputunu kullanır ve bize sınıflandırma sonucunu verir.



Genel Özet


MLP Head
Output
Transformer Encoder
CLS
1
2
3
n - 2
n - 1
n
Linear Projection of Flattened Patches
MLP
Batch Normalization
Multi-Head Attention
Batch Normalization
Embedded Patches
Fully Connected Layer
GeLU
Fully Connected Layer
Linear
Concat
head1
headh
Self-Attention
Self-Attention
Linear
Linear
Linear
Linear
Linear
Linear
Q
K
V
MatMul
SoftMax
Scale
MatMul
Q
K
V
Image by hongvin

İmplementasyonlar

Veri Seti : SceneParse150 (ADE20K)

Hazırlanan Dataset: https://huggingface.co/datasets/erent/scene_parse_5class
Hazırlanan Model: https://huggingface.co/erent/scene_parse_5class

Veri Seti : segments/sidewalk-semantic
Hazırlanan Model : https://huggingface.co/erent/sidewalk_semantic_4class

Veri Seti : Colorectal Cancer WSI
Hazırlanan Dataset : https://huggingface.co/datasets/erent/colarectal_WSI

Önisleme:

Veri Seti : SceneParse150 (ADE20K)

Aşağıdaki seçtiğim sınıflara ait olmayan sınıfları içeren görüntülerde belirtilen sınıfların dışındaki idlere 0 (arkaplan) idsi atandı. Ve sadece arkaplandan oluşan görüntüler datasetten çıkarıldı. Dataset boyutu 2000 satıra indirildi.

Class ID: 2, Name: windowpane
Class ID: 4, Name: person
Class ID: 5, Name: door
Class ID: 6, Name: table
Class ID: 10, Name: painting

Fine-Tune:

Veri Seti : SceneParse150 (ADE20K)

facebook/maskformer-swin-base-ade modelini temel alarak fine-tune ettim. Batch size'ı 32gb gpuyu tamamen kullanabilecek şekilde ayarladım çünkü batch size yükselmesi hem modelin sonuçlarını iyileştiriyor hemde daha hızlı train olmasını sağlıyor. Batch size'ı 14 olarak aldım. 10 epoch eğittim. lr=1e-4.

İOU Scores:

Class Name: cabinet, Score: 0.87
Class Name: door, Score: 0.92
Class Name: table, Score: 0.85
Class Name: curtain, Score: 0.82
Class Name: sofa, Score: 0.80

Inference :

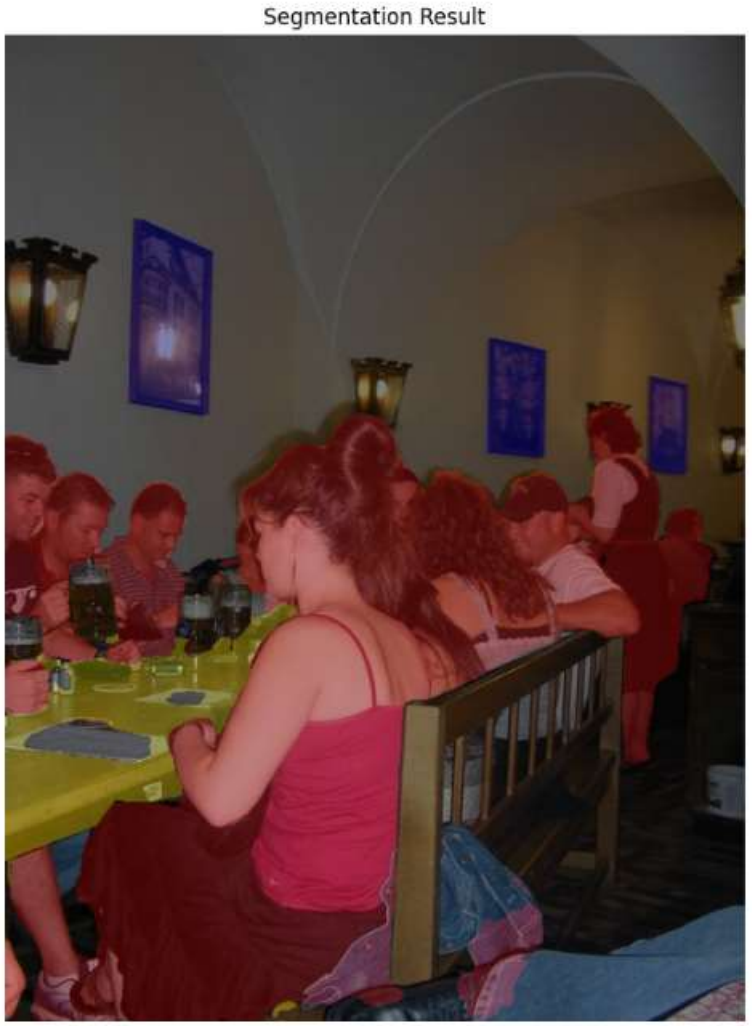
Veri Seti : SceneParse150 (ADE20K)

Segmentation Result

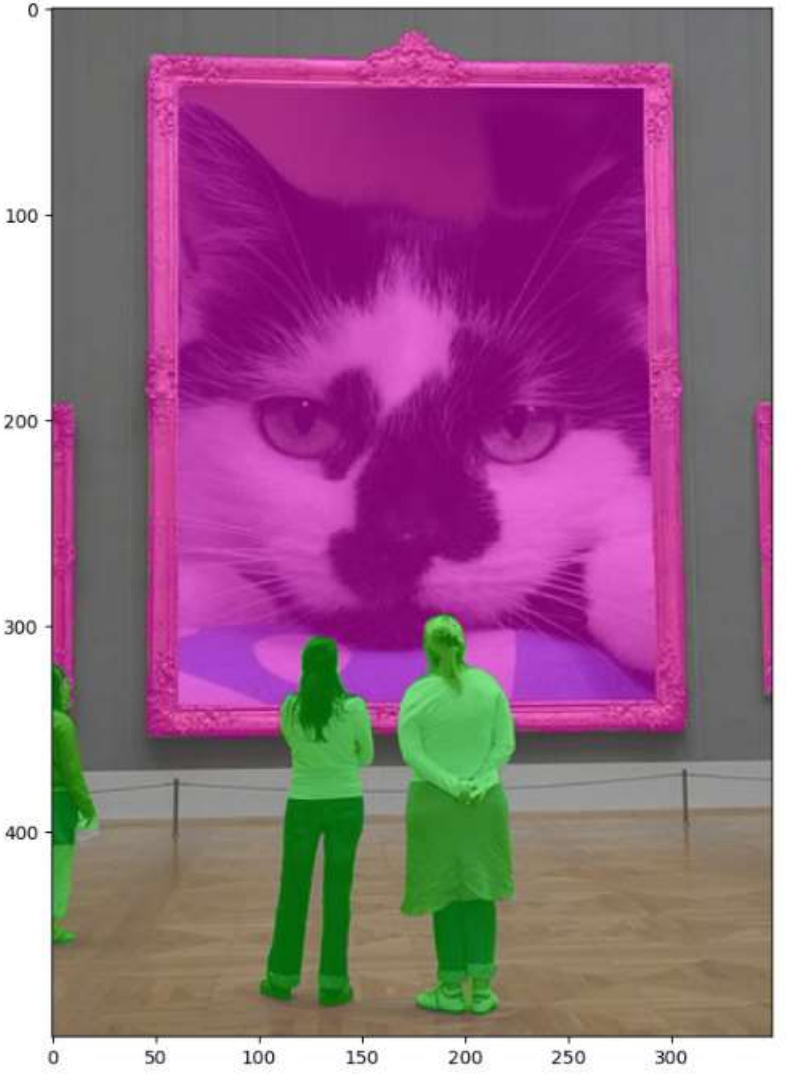
```
Class ID to Color and Name Mapping:
Class ID: 2, Color: cyan, Name: windowpane
Class ID: 4, Color: red, Name: person
Class ID: 5, Color: green, Name: door
Class ID: 6, Color: yellow, Name: table
Class ID: 10, Color: blue, Name: painting
```

Inference :

Veri Seti : SceneParse150 (ADE20K)




```
Class ID to Color and Name Mapping:
Class ID: 2, Color: cyan, Name: windowpane
Class ID: 4, Color: red, Name: person
Class ID: 5, Color: green, Name: door
Class ID: 6, Color: yellow, Name: table
Class ID: 10, Color: blue, Name: painting
```



Inference :

Veri Seti : SceneParse150 (ADE20K)


```
Class ID to Color and Name Mapping:
Class ID: 2, Color: cyan, Name: windowpane
Class ID: 4, Color: red, Name: person
Class ID: 5, Color: green, Name: door
Class ID: 6, Color: yellow, Name: table
Class ID: 10, Color: blue, Name: painting
```

Önisleme:

Veri Seti : segments/sidewalk-semantic

Aşağıdaki seçtiğim sınıflara ait olmayan sınıfları içeren görüntülerde belirtilen sınıfların dışındaki idlere 1 (arkaplan) idsi atandı. Görüntüler modele verilmeden önce data augmentation yapıldı. Normalizasyon, döndürme ve kırpma yapıldı.

Class ID: 8, Name: human-person
Class ID: 10, Name: vehicle-car
Class ID: 15, Name: vehicle-bicycle
Class ID: 26, Name: object-trafficsign

Fine-Tune:

Veri Seti : segments/sidewalk-semantic

facebook/maskformer-swin-base-ade modelini temel alarak fine-tune ettim. Batch size'ı 14 olarak aldım. 6 epoch eğittim. lr=5e-5.

Mean Accuracy: 0.89

İOU Scores:

Class Name: human-person, Score: 0.91
Class Name: vehicle-car, Score: 0.98
Class Name: vehicle-bicycle, Score: 0.84
Class Name: object-trafficsign, Score: 0.72

Inference :

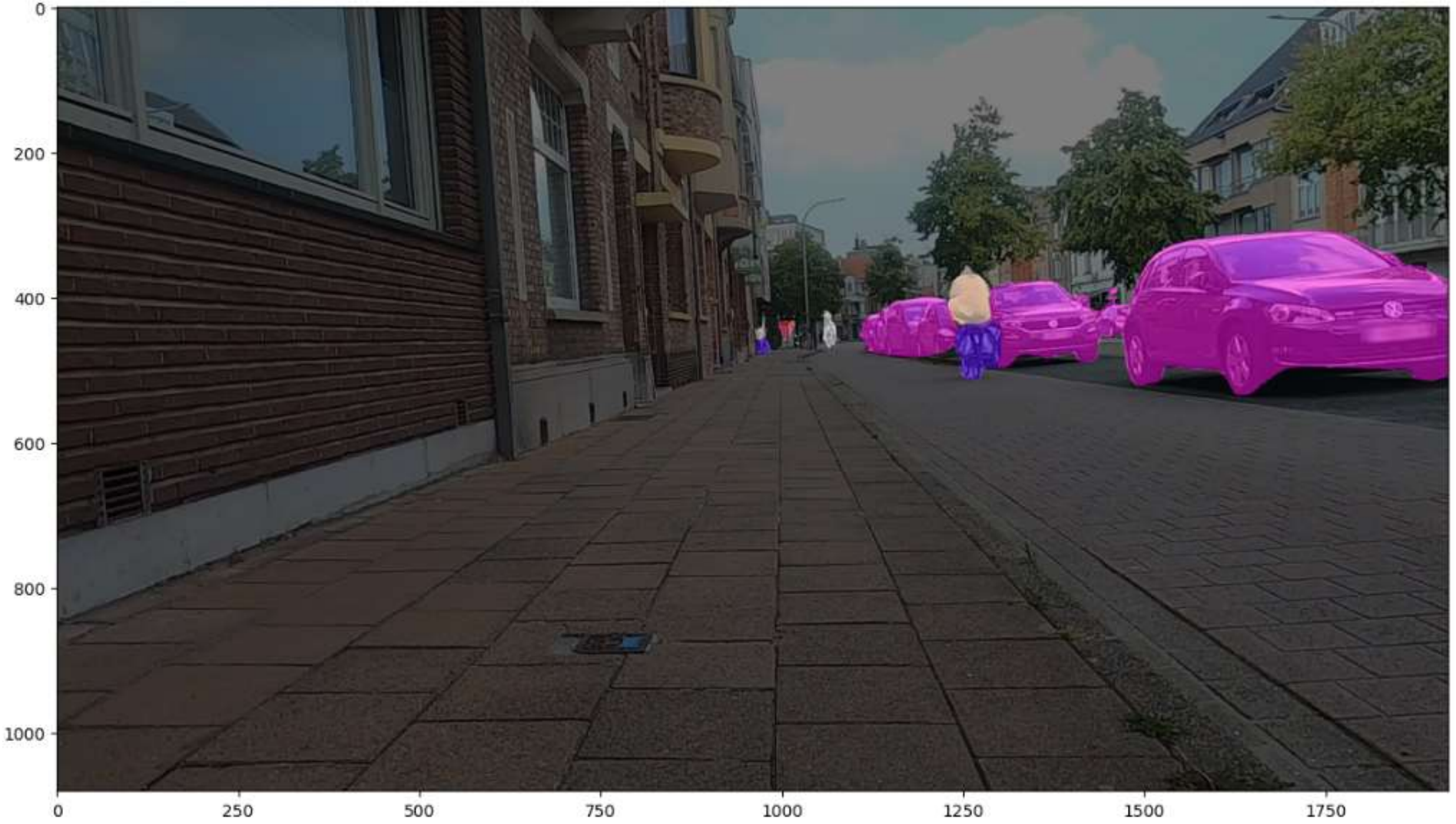
Veri Seti : segments/sidewalk-semantic

Segmentation Result

```
Class ID to Color and Name Mapping:
Class ID: 0, Color: [0, 0, 0], Name: unlabeled
Class ID: 8, Color: [128, 0, 0], Name: human-person
Class ID: 10, Color: [0, 128, 0], Name: vehicle-car
Class ID: 15, Color: [128, 128, 0], Name: vehicle-bicycle
Class ID: 26, Color: [0, 0, 128], Name: object-trafficsign
```

Inference :

Veri Seti : segments/sidewalk-semantic



Inference :

Veri Seti : segments/sidewalk-semantic

Segmentation Result

```
Class ID to Color and Name Mapping:
Class ID: 0, Color: [0, 0, 0], Name: unlabeled
Class ID: 8, Color: [128, 0, 0], Name: human-person
Class ID: 10, Color: [0, 128, 0], Name: vehicle-car
Class ID: 15, Color: [128, 128, 0], Name: vehicle-bicycle
Class ID: 26, Color: [0, 0, 128], Name: object-trafficsign
```

Inference :

Veri Seti : segments/sidewalk-semantic


```
Class ID to Color and Name Mapping:
Class ID: 0, Color: [0, 0, 0], Name: unlabeled
Class ID: 8, Color: [128, 0, 0], Name: human-person
Class ID: 10, Color: [0, 128, 0], Name: vehicle-car
Class ID: 15, Color: [128, 128, 0], Name: vehicle-bicycle
Class ID: 26, Color: [0, 0, 128], Name: object-trafficsign
```

Referanslar:

https://medium.com/@aliborji/a-unified-view-of-vision-transformer-and-mlp-mixer-4ccdab10c58c

https://www.youtube.com/watch?v=jzPbx9Y0vHg

https://www.youtube.com/watch?v=2V3Uduw
1zwQ

https://www.youtube.com/watch?v=eMlx5fFN
oYc&t=1045s

https://theaisummer.com/self-attention/

Teşekkürler

Eren Torlak