T.C.
KAHRAMANMARAŞ SÜTÇÜ İMAM ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GPU ile KUMAŞ HATA TESPİTİ ve SINIFLANDIRMASI

HASAN GÜLER

YÜKSEK LİSANS TEZİ

BİLGİSAYAR ve ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM DALI

KAHRAMANMARAŞ 2013

T.C.
KAHRAMANMARAŞ SÜTÇÜ İMAM ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GPU ile KUMAŞ HATA TESPİTİ ve SINIFLANDIRMASI

HASAN GÜLER

Bu tez,
Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalında
YÜKSEK LİSANS
derecesi için hazırlanmıştır.

KAHRAMANMARAŞ 2012

Kahramanmaraş Sütçü İmam Üniversitesi Fen Bilimleri Enstitüsü öğrencisi Hasan GÜLER tarafından hazırlanan "GPU ile KUMAŞ HATA TESPİTİ ve SINIFLANDIRMASI" adlı bu tez, jürimiz tarafından 13 / 12 / 2012 tarihinde oy birliği ile Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Doç. Dr. Mahit GÜNEŞ (DANIŞMAN) ……
Elektrik-Elektronik Mühendisliği, KSÜ

Yrd. Doç. Dr. MUSTAFA ŞEKKELİ ……

Elektrik-Elektronik Mühendisliği, KSÜ

Yrd. Doç. Dr. ORHAN ERCAN ……

Bilgisayar ve Öğretim Teknolojileri Eğitimi, KSÜ

Yukarıdaki imzaların adı geçen öğretim üyelerine ait olduğunu onaylarım.

Prof. Dr. M. Hakkı ALMA ……

Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Hasan GÜLER

Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

GPU ile KUMAŞ HATA TESPİTİ ve SINIFLANDIRMASI

ÖZET

Kumaş hatalarını tespit etmeye yönelik kalite kontrolü göz muayenesi yöntemi ile gerçekleştirildiğinde, zaman kaybı ve maliyet artışına neden olmaktadır. Son yıllardaki teknolojik gelişmeler bilgisayarlı görme sistemlerinin endüstriyel alanlarda kullanımına yönelik çalışmaları arttırmıştır. Bu tezde, örüntü tanıma yöntemiyle otomatik olarak kumaş hatalarını tespit edebilen ve sınıflandırmasını yapabilen bir sistem geliştirilmiştir.

Görüntü işleme yöntemi olarak Ayrık Dalgacık Dönüşümü (ADD) metodu kullanılmıştır. Sınıflandırma için ise başarılı sınıflandırma performansına sahip ileri beslemeli ve geri yayılımlı çok katmanlı yapay sinir ağı (YSA) algoritması kullanılmıştır. Kumaş hata tespiti ve sınıflandırması için geliştirilen uygulama, Merkezi İşlem Birimi (CPU) ve Grafik İşlem Birimi (GPU) üzerinde çalıştırılarak hız bakımından performans değerlendirmesi araştırılmıştır. CPU ve CPU'da çalıştırılmak üzere aynı algoritmalara sahip iki adet uygulama geliştirilmiştir. GPU performansını test etmek için NVIDIA GTX 480 ekran kartı ve bu ekran kartının programlanabilmesini sağlayan yazılımlar kullanılmıştır.

GPU ile elde edilen sonuçlar CPU'ya göre karşılaştırıldığında; GPU'nun öznitelik çıkarımı için 2, öznitelik vektörleri seçimi ve normalizasyon için 1,5, sınıflandırma için 10 ve toplam cevap süresi için ise yaklaşık 6 kat daha hızlı performansa sahip olduğu görülmüştür.

Anahtar Sözcükler: *CUDA, GPU, Dalgacık Dönüşümü, Yapay Sinir Ağı*

FABRIC DEFECT DETECTION AND CLASSIFICATION WITH GPU

SUMMARY

Fabric quality control is still done with human vision, so it causes both time consuming and cost increasing in production. In recent years, technologic improvements increased studies related to the use of computer vision systems in industrial fields. In this study, a system was developed to detect and classify automatically fabric faults by pattern recognition method.

As image processing tool DWT method was used. For classification, a multilayer feedforward neural network with backpropagation algorithm, which has good classification performance, was used. To improve application runtime performance, programmable GPU technology having rapidly developed in years was used. The application developed for fabric defect detection and classification was evaluated on CPU and GPU in terms of runtime performance. Two applications having the same algorithm were developed to run on CPU and GPU. To test GPU runtime performance for application NVIDIA GTX 480 graphic card and the software which enable to program or code GPU was used

When GPU performance results compared with CPU, it was seen that GPU has 2 times for feature extraction, 1.5 times for feature selection and normalization, 10 times for classification and 6 times for total response of application faster performance.

Anahtar Sözcükler: *CUDA, GPU, Wavelet Transform, Neural Network*

TEŞEKKÜR

Yüksek lisans öğrenimim süresince danışmanlığımı yapan ve çalışmalarım sırasında desteğini esirgemeyen, Sayın Doç. Dr. Mahit GÜNEŞ' e çok teşekkür ederim.

Hasan GÜLER

İÇİNDEKİLER

Sayfa No

SİMGELER ve KISALTMALAR DİZİNİ

CPU : Central Processing Unit

GPU : Graphic Processing Unit

CUDA : Compute Unified Device Architecture

NVCC : NVIDIA C Derleyicisi

DD :Dalgacık Dönüşümü

ADD : Ayrık Dalgacık Dönüşümü

SDD : Sürekli Dalgacık Dönüşümü

MSE : Mean Squared Error

SSE : Sum Squared Error

RMS : Root Mean Squared Error

YSA : Yapay Sinir Ağı

MLP : Multi Layer Perseptron

LVQ : Learning Vector Quantization

NN : Neural Network

FFNN : Feed Forward Neural Network

ŞEKİLLER DİZİNİ

Sayfa No

TABLOLAR DİZİNİ

Sayfa No

1. GİRİŞ

Tekstil endüstrisinde, kumaş hataları tespiti kalite kontrolü için önemli bir rol oynamaktadır. Çünkü tekstil endüstrisinde oluşan hataların büyük bir çoğunluğu kumaş hatalarından oluşmaktadır. Oluşan bu hataları önleyici tedbirler alınabilmesi için hataların tespiti ve tanımlanması bir zorunluluk teşkil etmektedir. Birçok kumaş hata tipi mevcuttur ve bu hataların çoğu da makine arızaları nedeniyle oluşur. Dokuma esnasında atkı yönünde yâda çözgü yönünde meydana gelen; iplik kopması veya iplik kaçması gibi durumlar hatalara neden olur. Bazı hatalar ise iplikten kaynaklı (ipliğin üretimi sırasında veya daha sonra oluşmuş iplik kusurları) yâda makineden kaynaklı yağ damlatma gibi sebeplerden meydana gelir (Sari-Sarraf ve Goddard, 1999).

1.1 Kumaşlarda Genel Hata Çeşitleri

Kumaş: Birbirlerine dik ve paralel konumda bulunan ipliklerin desene bağlı olarak birbirlerinin altından üstünden geçirilmesi ile oluşan kaplayıcı yüzeylerdir. Pamuk, yün, ipek, keten vb. maddelerden elde edilir.

Atkı, dokumacılıkta kumaşı oluşturan iki iplik sisteminden kumaşın enine doğru yani kumaş kenarına dik olarak yerleşenleri, **çözgü** ise, bu iki iplik sisteminden kumaşın kenarına paralel uzanan iplikleri tanımlar. Atkı iplikleri, çözgü iplikleriyle dik açılıdır. Bu iki iplik birlikte kumaşın dokusunu oluşturur. Çözgü iplikleri ile atkı ipliklerinin birbiri ile kesişmeleri belli kurallara bağlı olarak gerçekleşir, bu kurallar ise ilgili kumaşın örgü desenine göre değişir. Atkı ipliğinin rengi kumaşın zemin rengini çözgü ipliğinden daha çok etkiler.

Belirgin hata tipleri ve sınıflandırılması aşağıdaki gibidir;

- Çözgü yönündeki hatalar
    - Çözgü kaçığı
    - Çift çözgü
    - Gevşek Çözgü
    - Yüzen İplik
- Atkı yönündeki hatalar
    - Atkı Kaçığı
    - Gevşek Atkı

    - Çift Atkı
- Diğer hata tipleri
    - Şantuk Hatası
    - Düğümlenme Hatası
    - Yabancı Madde Hatası
    - Yağ Lekesi Hatası
    - Delik Hatası

1.1.1 Çözgü yönündeki hatalar

Çözgü kaçığı: Dokuma esnasında çözgü ipliğinin kopması ile oluşan hata tipidir. Çözgü ipliklerinden bir veya birkaçının koparak eksik kalması sonucunda atkı iplikleri, eksik olan çözgü iplikleriyle bağlantı yapamayacağından hata meydana gelir. Bu hata, doku yüzeyinde çözgü yönünde bir çizgi halinde kendini gösterir. Çözgü kaçığı kumaşın görünüşünü bozması yanında mukavemetini de düşürür. Otomatik tezgâhlarda çözgü ipliklerinin bazısı koparsa ve çözgü kontrol cihazı makineyi hemen durdurmazsa, kumaş üzerinde eksik iplikler nedeniyle çizgiler meydana gelir. Bu çizgiler, dokumacı tarafından görülüp tezgâh durdurulana kadar devam eder. Ayrıca, dokuma hazırlık dairesindeki hatalar yüzünden bobinlerdeki ipliklerden bazısı levende sarılırken biterse, dokumacı da buraya iplik ilavesini ihmal ederse birkaç metre uzunluğunda çizgiler oluşur.

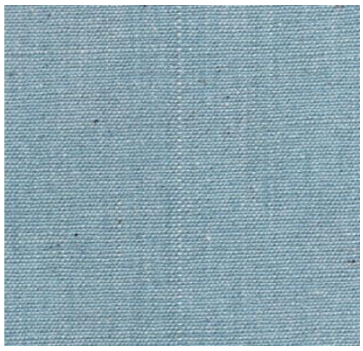
Çift Çözgü: Dokuma esnasında kumaşın normal çözgü ipliğinin yanında ekstra bir çözgü ipliği ile birlikte dokunmasıyla oluşan hata tipidir. Bütün dokumalarda görülebilen bir hatadır. Özellikle gevşek dokumalarda ve sentetik ipliklerden yapılan düz kumaşlarda kendini belli eder. Doku yüzeyinde yan yana iki çözgü ipliği meydana gelebileceği gibi, hafif kabarıklık yaparak kalın bir çözgü oluşturur. Kumaşta ışıklı kontrol cihazından bakıldığı zaman yan yana bulunan iki çözgü ipliğinin atkılarla ayı bağlantıyı yaptığı görülür.

Gevşek Çözgü: Bir çözgü ipliğinin diğerlerine nazaran daha gevşek dokunması ile oluşan hata tipidir. Gevşek çözgü, yeterli gerginlikte olmayan bir veya birkaç çözgü telidir. Bir kumaşta yanındaki normal çözgü ipliklerininkinden daha düşük bir gerginlikle dokunan çözgü ipliği veya çözgü iplikleri nedeniyle oluşan bir hatadır. Dokumanın yüzey düzgünlüğünü bozarak kabarık bir durum meydana getirir.

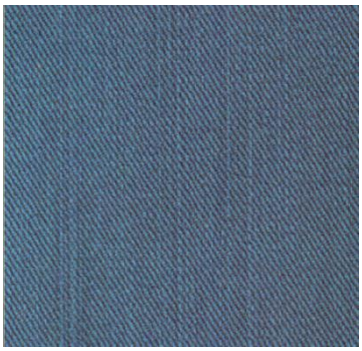
Yüzen İplik (atlama hatası) : Dokumada örgü içerisinde bulunması gereken çözgü ipliğinin belirli bir süreliğine kumaş yüzeyinde seyretmesi ve daha sonra örgüye tekrar katılması ile oluşan hata tipidir.

Çözgü yönündeki hata türlerine örnek resimler Şekil 1.1'de verilmiştir.

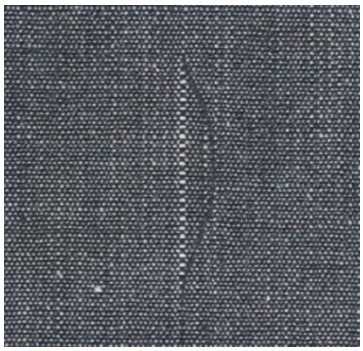
Çözgü kaçığı



Çift Çözgü



Gevşek Çözgü



Yüzen İplik

Şekil 1.1. Çözgü yönündeki hata tipleri

1.1.2 Atkı yönündeki hatalar

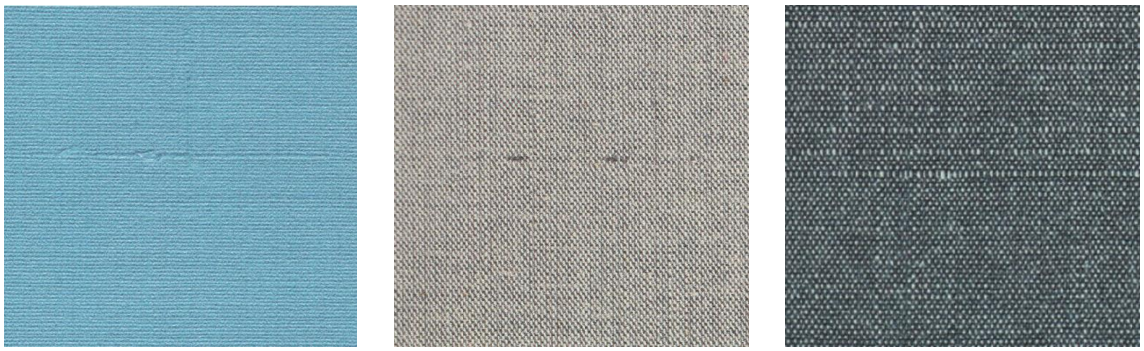
Atkı Kaçığı: Dokuma esnasında atkı ipliğinin kopması ile oluşan hata tipidir. Atkı kaçığı, dokuda bir ya da birkaç atkı telinin eksikliğidir. Bu hata atkı ipliklerinin kumaş eninde tamamen eksikliğinden kumaşta bir aralık oluşmasıdır. Kumaş genişliğinin tamamına atkı yerleştirilmemiştir. Atkı kaçığı tüm dokumalarda rastlanabilen bir hata olup şardonlama işlemi gören kumaşlarda belli olmamaktadır. Bu hata ince dokumalarda çok dikkati çeker. Eğer atkı kaçığı küçükse ve değeri yüksek kumaşlarda ham kontrol işleminde örme yoluyla giderilebilir. Fakat büyükse, dokuma enince devam ediyorsa maliyeti nedeniyle hiçbir işlem yapılamaz.

Gevşek Atkı: Bir atkı ipliğinin çözgü iplikleri arasından atılırken atkı taşıyıcı aparatının ipliği belirli gerginlikte kumaşa dâhil edememesinden kaynaklanır. Gevşek atkı, yeterli gerginlikte atılmayan bir ya da birkaç atkı telidir. Bir veya daha çok atkı ipliğinin

normal atkı ipliklerine nazaran, daha düşük yetersiz gerginlikte dokunmasından ileri gelen dokuma hatasıdır.

Çift atkı: Çift atkı, ağızlıktan çift atkı ipliğinin geçmesi ile oluşan hatadır. Dokuma sırasında aynı ağızlığa yanlışlıkla tek atkı ipliği yerine iki atkı ipliğinin yerleşmiş olmasından kaynaklanır. Bu hata kumaşın genişliğinde bir çizgi gibi görünür. Bu hatanın meydana geliş nedeni; atkı ağızlığın içinde koptuğu zaman yarım atkının sökülmeyip, atkı taşıyıcının yeniden atılmasıdır.

Atkı yönündeki hata türlerine örnek resimler Şekil 1.2'de verilmiştir.



Atkı kaçığı Gevşek atkı Çift atkı

Şekil 1.2. Atkı yönündeki hatalar

1.1.3 Diğer hatalar

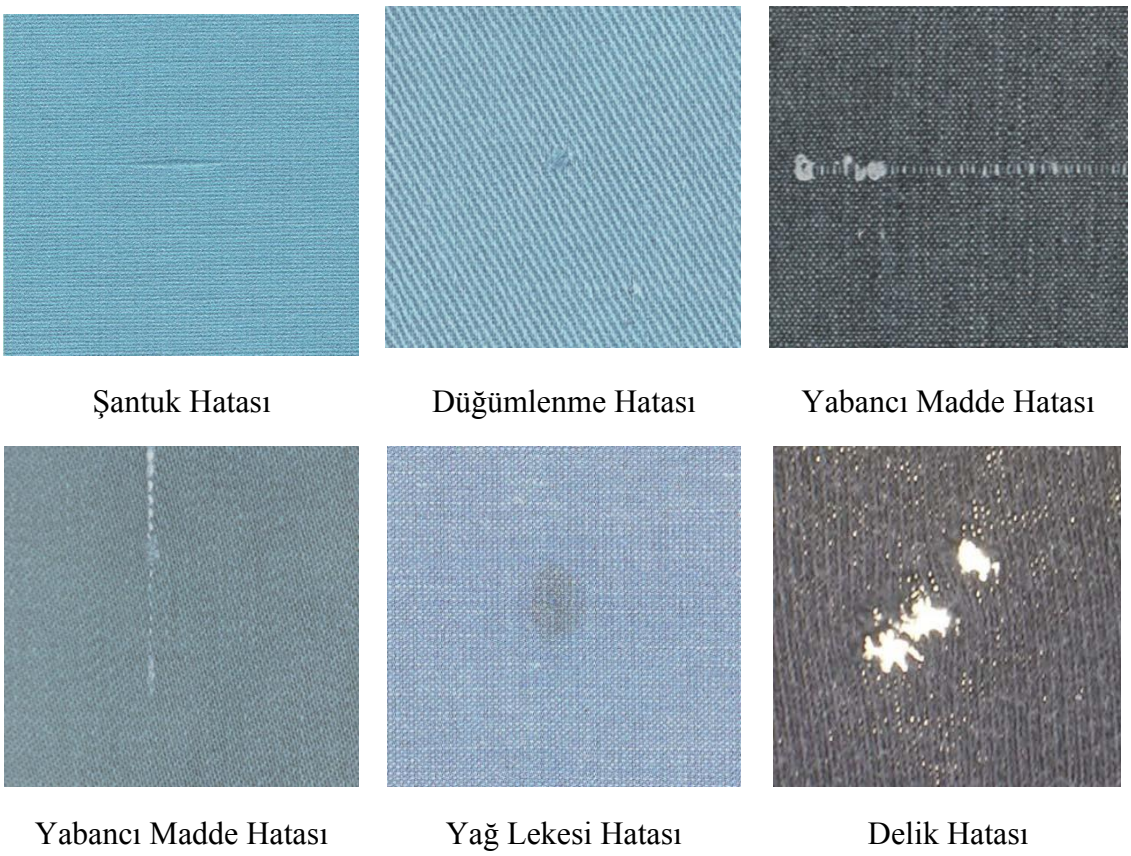
Şantuk Hatası: Şantuk; ipliğin belli bir bölümünün normal kalınlığından daha kalın olması durumudur. Dolayısıyla dokuma esnasında iplikte bu kalınlık farkından meydana gelen hata tipi de şantuk hatasıdır**.**

Düğümlenme Hatası: İpliklerinin birbirlerinde dolaşarak kumaş yüzeylerinde oluşturdukları hata tipidir. Dokumada kopan ipliklerin düğümlenmesinden kaynaklanan hatadır.

Yabancı Madde Hatası: Dokuma sırasında kumaşa dâhil olan bazı yabancı maddelerin (elyaf, iplik gibi) oluşturduğu hata tipidir. Dokuma esnasında, çözgü ipliğinde elyaf topaklarının oluşumudur. Düğümlenme veya çözgü ipliği kopmalarına yol açabilir ya da her iki hata birden oluşabilir. Atkı atma mekanizmasında toplanan elyafta kumaş içine geçerek balık, palamut hatalarını oluşturur veya atkı kopuşuna neden olabilir. Çözgü veya atkıda yabancı bir ipliğin bağlanmasından oluşan ve bir çizgi halinde görülen hatadır.

Yağ Lekesi ve Delik Hatası: Makine yağının kumaş ya da iplik üzerine damlasından meydana gelir. Delik hatası ise genelde kırılmış iğneden kaynaklanır.

Diğer hata türlerine örnek resimler Şekil 1.3'de verilmiştir.

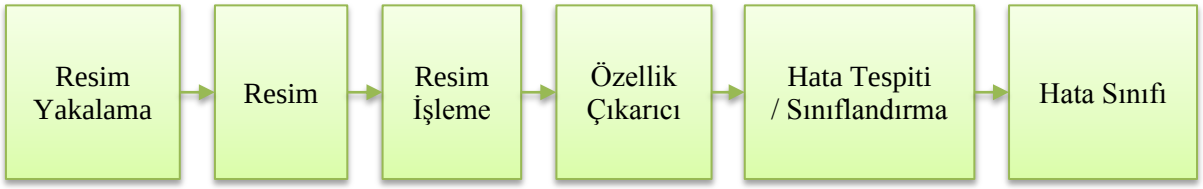


Şekil 1.3. Diğer hata tipleri

1.2 Kumaş hata tespiti ve sınıflandırma yöntemleri

Temel olarak çoğu hatalar kumaşın üretimi esnasında gerçekleştiği için kumaş hata tespitine yönelik en uygun çözüm de kumaşın üretim esnasında otomatik olarak kontrol edilmesi ile sağlanabilir. Fakat kumaş hata tespiti, insan kontrollü olarak ve ancak belirli düzeyde kumaş miktarı üretildikten sonra sağlanabilmektedir. Bu işlem üretilen kumaşın makineden alınıp rulolara sarıldıktan sonra hata tespit bölümüne gitmesi ile başlar ve bu bölümde hata tespiti gerçekleştirilir. Gözle yapılan kontroller ürünlerin değerlendirilmesinde iyi sonuçlar vermemektedir. Bu nedenle otomatik görsel tespit sistemleri insan kontrollü tespit sistemlerine karşılık iyi bir alternatif sayılır. Otomatik hata tespit ve tanıma sistemi; ürün kalitesini arttırma, tüketici ihtiyaçlarını karşılama ve üretim maliyetlerini azaltma açısından verimlilik sağlar (Tsai ve Huang, 2003).

Kumaş hataları tespitinde, bilgisayar donanımı ve yazılımlarındaki gelişmelere bağlı olarak, görüntü işleme, örüntü tanıma gibi yöntemler güvenilir, objektif ve kararlı performanslar sağlarlar. Bilgisayar kontrollü hata tespiti son yıllarda en önemli uygulama alanlarından biri haline gelmiştir. Görüntüsü alınan resimler üzerinde bilgisayar ortamında görüntü işleme teknikleri ile kumaş hataları tespiti, tanımlanması ve sınıflandırılması yapılabilir. Genel olarak otomatik hata tespit sistemi Şekil 1.4'te verilmiştir.



Şekil 1.4. Otomatik hata tespit sistemi

Kumaş hata tespit sistemi, öznitelik çıkarıcı ve hata tespit edici olmak üzere iki önemli bileşeni içermektedir. Öznitelik çıkarıcı bir nevi karar mekanizması gibidir yani tespitin doğruluğuna karar verir. Nedeni ise kumaş görüntüsü üzerinde hatalı bölge ile hatasız bölge arasındaki elde edilen özniteliklerin farklı olmasıdır. Hata tespit edici ise gelen kumaş özniteliklerinden hatanın varlığını bulmaya yardım eder.

Normal kumaş ile kumaş hatalarını ayırt edebilmek için, çoğu dokularda iki genel algısal yaklaşım, düzenlilik (regularity) ve yerel oryantasyon vardır. Düzenlilik yâda periyodiklik aslında periyodik bir örüntüden ibarettir (Chetverikov ve Hanbury, 2002). Yerel oryantasyon ise bir örüntüyü tanımlamadaki karakteristik yönler anlamına gelir. Kumaş görüntülerinin doğasında bu iki özellik de bulunabilir. Kumaş hataları düzenli kumaş örüntüsünün yerel bozulmasından kaynaklanır. Bu bozulmalar uzunluk, genişlik ve gri seviyedeki değişimlere bağlı homojen olmayan düzensizliklerden oluşur. (Chung-Feng ve Kuo, 2003).

Hata tespiti daha önce de bahsedildiği gibi öznitelik (özellik) çıkarımına bağlıdır. Bu bağlamda öznitelik çıkarım metotları dört farklı yaklaşım altında sınıflandırılabilir. Bunlar istatistiksel yaklaşımlar, spektral yaklaşımlar, model-tabanlı yaklaşımlar ve yapısal yaklaşımlardır (Mahajan ve ark., 2009).

1.3 İstatiksel Yaklaşımlar

İstatiksel yaklaşımlarda, bir resim çok sayıda alt resimlere bölünür. Her bir alt resim içerisinde, birinci dereceden istatistik yada yüksek dereceden istatistik uygulanarak

gri seviye istatistik bilgileri karakterize edilir. İstatiksel yaklaşımda, görüntü piksel değerlerinin uzaysal dağılımlarını ölçerek gerçekleştirilen bir yöntemdir. Bu yaklaşımdaki önemli bir varsayım hatasız bölgelerdeki istatistiksel verilerin durgunluğudur ve bu bölgeler taraması yapılacak görüntünün büyük bir bölümünü oluşturur. İstatiksel yöntemler yerel yâda bölgesel özellikleri tanımlayan piksel sayılarına bağlı olarak birinci dereceden, ikinci dereceden ve yüksek dereceden istatistiki olarak sınıflandırılabilir. Birinci dereceden istatistik, görüntü pikselleri arasındaki uzaysal etkileşimi değerlendirmeden bireysel piksel değerlerinin ortalama, varyans, medyan çarpıklık, basıklık gibi özelliklerini tahmin eder. Diğer taraftan ikinci ve yüksek dereceden istatistik ise belirli konumlardaki iki veya daha fazla piksellerin bağıl özelliklerini tahmin eder ve homojenite, entropi gibi değerler açısından gri seviye ortak oluşum matrisi oluşturur (Chen ve ark., 1993, Jain ve ark.,2000). İstatistiksel yaklaşımlarda sadece büyük hata bölgeleri tespit edilebilir. Nedeni ise istatistiksel yaklaşımlarda nadiren yerel istatistiki bilgi elde edilir bu da küçük hata bölgeleri için ayırt edici bir bilgi vermez. Diğer taraftan yüksek dereceden istatistiksel yaklaşım resim tanımına yönelik doğru sonuç vermesine rağmen işlem süresi açısından da oldukça zaman alıcıdır. Bu yaklaşımlar, doku primitifleri ve bu primitiflerin uzaysal düzenlemesi ile karakterize edilir (Vilnrotter ve ark., 1986)

Bu yaklaşımların öncelikli amaçları önce doku primitiflerini çıkarmak sonra da uzaysal düzenleme kurallarını genelleme yâda modellemedir. Doku primitiflerinin yüksek derece periyodikliği, hata tespiti için uzaysal özelliklerin kullanımına izin verir. Fakat düzensiz dokuya sahip görüntüler, primitifler açısından tanımlanamaz. Aynı zamanda böyle görüntülerde gri seviyelerin dağılımı gibi yer değişim kuralları rastlantısaldır. Bu nedenle de spektral yaklaşımlar düzensiz dokuya sahip görüntüler için hata tespiti açısından uygun değildir. Literatürde bulunan birçok çalışmada, tekdüze dokulu materyallerdeki hata tespiti için uzaysal frekans alanı ve frekans özelliklerin kullanıldığı görülmüştür (Ngana ve ark., 2011)

1.4 Spektral Yaklaşım

Spektral alan yaklaşımlarında resmin, uzaysal domaindeki gri seviye değerleri spektral filtreleme, Gabor filtreleme, Fourier dönüşümü, dalgacık dönüşümü gibi işlemlerle farklı domainlere dönüştürülür. Dönüştürülmüş resimlerden de öznitelik verileri elde edilir. Bu dönüşüm teknikleri doku bölütlemesinde, sınıflandırmasında ve kumaş hataları tespitinde geniş çapta kullanılır.

Spektral filtreleme teknikleri, bir grup spektral maskeler üzerinde daha büyük bir alan oluşturarak kumaş hatalarını karakterize eder. Fourier dönüşüm yöntemleri kumaş dokusunun periyodik yapısını analiz eder ve hataların varlığını güç spektrumundan tespit eder. Dalgacık dönüşümü ve Gabor filtreleme, çoklu ölçek (skala) ve oryantasyonda kumaş resimlerinin konumlandırılmış spektral frekans analizini olanaklı kılar. Bu nedenle de hataları tespit etmekte resmin orijinal çözünürlüğündeki durumundan daha yeteneklidir (Unser, 1995).

1.5 Model-tabanlı yaklaşımlar

Model tabanlı yaklaşımlarda, hatasız kumaş resmi çok sayıda farklı pencerelere bölünür ve Gaussian Markov Rastgele Alanı ile modellenir. Hatalı bölgeler Hatasız bölgeler gibi değildir ve önemli ölçüde yüksek değerler üretir ve bir eşikleme uygulanarak hata tanımlanabilir. Bu yaklaşımın dezavantajı küçük hataları tespit etmesindeki zorluktur. Çünkü hatasız ve hatalı kumaş resimleri arasındaki uzaysal ilişkiyi temsil etmek için model parametrelerinin tahmini yeterince geniş değildir (Duda ve ark., 2001).

1.6 Yapısal yaklaşım

Yapısal yaklaşımda öznitelikler gri seviye yerleşimi ve hizalanmasına bağlı olarak elde edilir. Kumaş resimlerinde düzenli örüntüler bulunur. Bu örüntüler kendi primitifleri ile tekrar üretilir. Hatasız kumaş resimlerinin doku özellikleri ile kıyaslandığında, hatalı bölgeler tespit edilebilir çünkü hatalı bölgeler kendine özgün primitif yapılar içerir. Bu yaklaşımın olumlu yönü sabit bir pencere boyutu ile sınırlı değildir çünkü primitiflerin ebatları farklı kumaş hatalarına göre çeşitlilik gösterir. Olumsuz yönü ise gri seviye yapısı açısından, hatasız kumaş primitiflerinden farklı olması gereken hatalı bölge primitiflerini seçmedeki sınırlılığıdır (Ngana ve ark., 2011).

Yukarıda bahsi edilen görüntü işleme ve örüntü tanıma teknikleri için sınıflandırıcılara ihtiyaç vardır. Sınıflandırıcılar En yakın komşular, yapay sinir ağları ve destek vektör makinaları olmak üzere üç ana grupta toplanabilir. Yapay sinir ağları parametrik olmayan doğası karmaşık karar bölgelerini tanımlama gibi yeteneklerinden ötürü hata tespiti için en hızlı ve en esnek sınıflandırıcılardan birisi olarak düşünülebilir (Kumar, 2003)

2. ÖNCEKİ ÇALIŞMALAR

Bu bölümde örüntü tanıma, kumaş hata tespiti ve hata sınıflandırması üzerine, daha önce yapılan çalışmalar ve bu çalışmalarda kullanılan yöntemler detaylı olarak incelenmiştir.

2.1 İstatistiksel Yaklaşımlar ile Kumaş Hata Tespiti

2.1.1 Otokorelasyon ile kumaş hata tespiti

Wood (1990), halı dokularında ki hata tespitine yönelik kaynak(girdi) resimlerinden spektral frekans içeriğini elde etmede Fourier güç spektrumunu kullandı. Hatasız bir halı dokusu resmi için, Fourier spektrum ipliğin periyodik olarak tekrarlandığı güçlü ve eşit dağılmış tepeleri gösterir. Bu nedenle, bir otokorelasyon fonksiyonu uygulayarak, doku kalınlığı ve tekrarlanan örüntü için düzenlilik ölçümü yaptı. Örüntü değişimini tanımlayabilmek için de çapraz-korelasyon uyguladı.

2.1.2 Ortak oluşum matrisi ile kumaş hata tespiti

Ortak oluşum matrisi üzerine ilk çalışmaları Haralick ve ark. (1973) başlatmıştır. Enerji, entropi, kontrast, homojenite ve korelasyon gibi doku özellikleri ortak oluşum matrislerinden elde edilir. Bu değerler birçok hata tespiti için kullanılmıştır. Conners ve ark. (1983), ahşap yüzeyindeki 9 tip hata tespiti için bu 6 özelliği kullanmıştır. Rösler (1992), ortak oluşum matris özelliklerini kullanarak gerçek bir hata tespit sistemi geliştirmiş ve 1 mm^2 gibi küçük ebatlardaki hataların %95'inin tespit edildiğini belirtmiştir. Bodnarova ve ark. (1997) hata tespitinde bu konuyu optimal yer değiştirme vektörü ile test etmişleridir. Ortak oluşum vektörü birçok zorluk barındırmaktadır. Genel anlamda görünen şu ki yer değiştirmeyi optimize eden kabul görmüş bir çözüm mevcut değildir. Gri seviye verileri, ortak oluşum matrislerinin kullanılabilir düzeye getirilebilmesi için azaltılır. Bir yer değiştirme vektörü için çok fazla sayıda özellik verileri hesaplanabilir. Gerçek zamanlı hata tespiti için bu yöntem hesaplama açısından oldukça maliyetli olmasına karşın sonuçlarında ki doğruluktan kaynaklı çoğu araştırmalarda kullanılmıştır. Latif-Amet ve ark., (1998;2000) önerdikleri alt bant ortak oluşum matrisi yöntemi ile 36 adet 256×256 piksel ebadındaki kumaş resmi üzerinden % 90 başarı sağlamışlardır.

2.1.3 Morfolojik işlemleri ile kumaş hata tespiti

Zhang ve Bresee (1995), kumaş hataları tespiti için morfolojik işlemleri detaylandırdı. Özellikle kumaşın iyi ve hatanın küçük ebatta olduğu durumlarda uzaysal olarak filtre edilmiş kumaş görüntülerinde morfolojik hata tespitleri iyi sonuçlar vermiştir. Çalışmalarında morfolojik işlemleri sadece periyodik görüntüler üzerinde uygulamışlardır. Mak ve ark. (2005; 2009) tarafından tasarlanan optimal morfolojik filtre ile 78 adet 256×256 piksel ebadındaki kumaş resmi üzerinden yapılan kumaş hata tespitinde %95 ile %98 arasında başarı sağlanmıştır.

2.1.4 Fraktal analiz ile kumaş hata tespiti

İlk olarak fraktal tabanlı analiz Pentland tarafından tanıtıldı (Pentland, 1984). Conci ve Proença (1998), fraktal analiz yöntemi ile hatalı alanların belirlenmesinde 256×256 piksel ebadındaki 75 resim kümesi ve yine aynı ebattaki 80 resim kümesi üzerinde yaptıkları çalışmalarında 8 tip hata için %96 başarı elde etmişlerdir. Bu ve ark. (2009), çalışmalarında 14378 hatasız ve 3222 hatalı 256×256 piksel ebadındaki kumaş resimleri üzerinde fraktal özellikleri kullanarak %94 ile %99 arasında başarı sağlamışlardır.

2.2 Spektral Yaklaşım ile Kumaş Hata Tespiti

2.2.1 Fourier dönüşümü ile kumaş hata tespiti

Fourier dönüşümü görüntülerin frekans bileşenlerini karakterize eder. Periyodik özellikler frekans bileşenlerinin büyüklüğüyle gözlenebilir. Doku örüntüleri için spektrumda yüksek enerji çıkışlarının yoğunluğu kolayca ayırt edilebilir.

Wood (1990), halı dokularında ki hata tespitine yönelik kaynak(girdi) resimlerinden uzaysal frekans içeriğini elde etmede Fourier güç spektrumunu kullandı. Hatasız bir halı dokusu resmi için, Fourier spektrum ipliğin periyodik olarak tekrarlandığı güçlü ve eşit dağılmış tepeleri gösterir. Bu nedenle, bir otokorelasyon fonksiyonu uygulayarak, doku kalınlığı ve tekrarlanan örüntü için düzenlilik ölçümü yaptı. Örüntü değişimini tanımlayabilmek için de çapraz korelasyon uyguladı. Ravandi ve Toriumi (1995), Fourier dönüşümü kullanarak atkı ve çözgü yönündeki ipliklerin sıklığı, tüm kumaş üzerindeki iplik yoğunluğu gibi karakteristik özellikleri incelemiştir. Hoffer ve ark. (1996) özellik çıkarımı için optik Fourier dönüşümü kullandı. Sonrada tezgâh üzerinde hata tespiti ve sınıflandırması yapabilmek için 3 katmanlı bir yapay sinir ağı geliştirdi. Kumaş resmindeki

herhangi bir hata referans spektrum ile belirgin farklılıklar gösterdiğinden dolayı, Castellini ve ark.(1996) ve Ciamberlini ve ark. (1996), benzer bir çalışma yaparak bant üzerinde tekstil kumaşı denetim sisteminde optik Fourier dönüşümünü uyguladı. Atkı ve çözgü yönünde 2 boyutlu çerçevelenmiş uzaysal frekanslara bağlı, düzenli bir periyodik örüntü, optik Fourier dönüşümü ile yatay ve dikey konumlu bir çift tepeler serisi meydana getirir. Bazı farklı tekniklerle, düzensizlikler yani hatalar bitişik tepeler arasındaki ışık yoğunluğun yükselişi bir varyasyon üretir. Bu yaklaşımlar, tezgâh makinesinin titreşimine ve elektriksel bozukluklara hassas iken küçük hatalara karşı hassas değildir.

Campbell ve ark. (1997), ayrık Fourier dönüşümünü 11 katmanlı yapay sinir ağı ile birlikte kot kumaşı hata tespitinde uyguladı. Hata olasılığının önceden tahminine yönelik geçerli bir çözüm sağlamak için önemli testler gerçekleştirdi. Campbell ve ark. (1997), Fourier dönüşümünden özellik vektörlerini elde ederek maksimum olasılık sınıflandırıcısı ile kusur (hata) tespiti gerçekleştirmeyi başarmıştır. Escofet ve ark. (1998), aşınmaya karşı kumaş direncini değerlendirmek amacıyla Fourier spektrumun açısal korelasyonunu kullanmıştır.

Özdemir ve ark.,(1998), kumaş hata tespitinde Fourier dönüşümünü kullanmış K-L dönüşümü ve örgü filtreleme yöntemleri ile karşılaştırmışlardır. 256×256 piksel ebadındaki gri seviyeli 9 hatalı kumaş resim örneklerini 32×32 piksel pencerelere ayırarak gerçekleştirmiş oldukları FFT tabanlı yaklaşımlarında tespit başarı oranı %87,5 olarak görülmekte ve Markov rastgele alan yaklaşımının daha başarılı ve hızlı olduğu belirtilmektedir.

Tsai ve Hsieh (1999), kumaş hatası tespitine yönelik Fourier dönüşümü kullanarak küresel bir resim yenileme şeması sundu. Görüntülerdeki satır örüntüleri hata olarak varsayılmıştır. Bir boyutlu Hough dönüşümü ile uzaysal alan resmindeki herhangi bir yönlü dokuların çizgi örüntüleri, Fourier alan resmindeki yüksek-frekans bileşenlerinin tespit edilip değerlerinin sıfırlanması ve son olarak da tekrar uzaysal alan resmine dönüştürülmesi yani ters Fourier dönüşümü ile ortadan kaldırılır. Böylece orijinal görüntü ile yeniden oluşturulmuş görüntü arasındaki fark potansiyel hata olarak düşünülmüştür Orijinal resimdeki hatasız bölgeler hemen hemen aynı gri seviyeyi gösterirken, hatalı bölgeler belirgin bir şekildedir.

Chan ve Pang (2000), kabartmalı yani fitilli kumaş üzerinde hata tespiti için 3 boyutlu Fourier frekans spektrumundan seçilen 2 merkezi uzaysal frekans spektrumu

kullanmayı önerdi. Çift iplik, iplik kaçığı, dokuma ve iplik yoğunluğu gibi dört farklı hata tipini ayırt etmek için Fourier spektrumdaki yatay ve dikey frekans bileşenlerinden özellik çıkarımı yaptılar. Merkezi uzaysal frekans spektrumu tespit için bilgisayar hesaplama zamanı açısından avantaj sağladı. Çalışmalarını gerçek kumaş örnekleri üzerinden 4 tip sınıflandırma için yapmışlardır.

Chiu ve ark. (2002), kumaş hatalarını tespit için fraksiyonel Brownian devinim modeline dayalı Fourier alanı maksimum olasılık tahmincisini kullandı. 128×128 piksel ebatlarındaki 4 hatalı kumaş resimleri bu model ile başarılı bir şekilde tespit edilmiştir. Bu yöntem resmin tekrar boyutlandırılması, döndürülmesi, konum değişimi, gri seviye değişimi gibi geometrik dönüşümlere göre değişiklik göstermez.

Tsai ve Huang (2003), Fourier dönüşümü ile küresel resim yenileme şemasını kullandılar. Resim yenileme esnasında optimum yarıçap ayarı yaparak ters Fourier dönüşümü ile herhangi bir istatistiki dokunun periyodik ve tekrar eden örüntülerini silmeye çalıştılar. Orijinal resimdeki homojen bölgeler hemen hemen aynı gri seviyeyi gösterirken, hatalı bölgeler belirgin bir şekildedir.

Baştürk ve ark. (2005), çalışmalarında 256×256 boyutlarında ve 8-bit gri seviyeli gerçek kumaş resimleri üzerinde Fourier dönüşümü, eşikleme ve Gabor süzgeçlerini kullanarak hata tespiti gerçekleştirmiştir. 9 adet hatanın tespit edildiği gösterilmektedir.

Sengottuvelan ve ark. (2008), Fourier dönüşümüne dayalı bir yaklaşım sundular. Çalışmalarında resim uzayındaki ve frekans uzayındaki kumaş yapısı arasındaki ilişkiyi incelmişlerdir. Üç boyutlu frekans spektrumundan 2 merkezi uzaysal frekans spektrumunu kullandılar. Gerçek örnekler üzerinden 7 önemli karakteristik parametreyi merkezi uzaysal frekans spektrumdan elde ederek 3 tip hata türü için sınıflandırma yapmışlardır.

Zhao ve ark. (2008), hata tespiti için yeni bir yaklaşım sundu. Hızlı Fourier dönüşümü ve öz-adaptif güç spektrum ayrıştırmasını uygulayarak spektrumun bölgesel enerjilerinden ortalamalarını ve standart sapmalarını kumaş öznitelikleri olarak elde etti. Kumaş hataları tanımada önerilen yöntemin çevrimiçi tespit de yüksek performans sağladığı belirtilmiştir.

Jayashree ve Subbaraman (2011), Fourier güç spektrumu toplamı ile morfolojik filtre tasarlayarak kumaş periyodiklik özniteliklerini elde etmiştir. Çalışmalarında gri kabartmalı kumaş hatalarını tespit etmişlerdir. Önerilen yöntem ile gerçekleştirilen hata tespiti için yüksek başarı oranı gözlemlenmiştir.

2.2.2 Gabor dönüşümü ile kumaş hata tespiti

Fourier dönüşümü, sinyaldeki global frekans içeriğinin bir analizidir. Hatalı bölgelerin konumları hakkında bilgi verme yeteneğine sahip değildir. Eğer pencerelenmiş Fourier fonksiyonu Gaussian ise o zaman Gabor dönüşümü olur. Uzaysal ve frekans alanında optimal konumlama sağlar (Daugman,1980).

Jain ve Farrokhnia (1991), Gabor dönüşümünü radyal uzaysal frekans değişiminin diyadik kapsaması ile kullanarak hata tespiti ve sınıflandırma yapmıştır. Escofet ve ark. (1998), dört frekans seviyeli ve dört oryantasyonlu 4×4. Gabor filtresini kumaş hatası tespiti için uygulamıştır. Gabor dönüşümün uygulamasında piramit yapı kullanmıştır. Tsai ve Wu (2000), enerji üretmek için spesifik bir Gabor filtresi ile filtre edilen kumaş resmine yönelik bir Gabor filtre şeması uygulamıştır. Hatalı bölgeler için, önemli derecede büyük enerji değerleri üretirlerken hatasız bölgeler için böyle değildir.

Kumar ve Pang (2002), bir sınıf kumaş hatası için danışmanlı hata tespit yaklaşımı altında Gabor dalgacık özniteliklerini seçti ve danışmansız kumaş hatası tespiti için de çoklu-iş parçacığı filtreleme şeması içeresinde öz-benzer Gabor fonksiyonlarını kullanmayı önerdiler. Aynı zamanda, Gabor fonksiyonunun sadece sanal (imajinari) kısmı ile tespiti için düşük hesaplama önerisini sundular.

Bodnarova (2000; 2002), homojen olarak dokunmuş kumaşlarda hata tespite yönelik iki boyutlu optimal bir Gabor filtresi tanıttırmıştır. Gabor fonksiyonun parametreleri, hatalı dokunun farklı bir frekans yanıtı vereceği Fisher değer fonksiyonunun minimizasyonu ile optimize edilmiştir.

Tsai ve Lin (2002), iki boyutlu Gabor filtreleme şemasını iki boyutlu gri seviyeli resmi bir boyutlu örüntü şekline dönüştürerek hataları tespit için bir boyutlu Gabor filtresi geliştirdi. Hesaplama karmaşıklığını azaltmak amacıyla böyle bir yaklaşım geliştirdi.

Shu ve Tan. (2004), çoklu iş parçacığı ve çoklu skalaya bağlı Gabor filtreleme ile otomatik tespit için bir yöntem geliştirmişlerdir. Bu yöntem farklı frekans ve oryantasyonlardaki Gabor filtre bankası konvolusyonun enerji yanıtına bağlıdır. Uygulanan birçok deneysel sonuçlar bu yöntemin verimliliğini ortaya koymaktadır.

Ogata ve ark. (2005), kabartmalı kumaş hatası tespiti için elektromanyetik dalga kalkanını görüntüleyecek, plazma görüntü panelinin arayüzü ile yeni bir resim görüntüleme tekniği önerdiler. Küresel hataları tespit etmek için 2 boyutlu ayrık Fourier dönüşümünü, lokal hataları bölümleme için ise optimal Gabor Filtrelerini uyguladılar.

Hou ve Parker (2005), Gabor dalgacık özellikleri ile Destek Vektör makinesinin sınıflandırma yaklaşımını kullanarak dokulu yüzeylerde hata tespiti için bir yöntem araştırmışlardır. Özellik çıkarımında hesaplama maliyetini azaltmak için Gabor dalgacık süzgeçlerinin tüm filtrelerini kullanmaktansa, mantıklı ve kabul edilebilir bir tespit oranı yakalayana kadar adaptif bir filtre seçimi gerçekleştirilmiştir. Deneysel sonuçlar hata tespitinde bu yöntemin daha başarılı sonuçlar verdiğini göstermektedir.

Arivazhagan ve ark. (2005), Gabor dönüşümünü hata tespitinde kullanılmıştır. Bu çalışmada İnsan görmesini taklit eden Gabor filtre şeması uygulanmış ve sonuçlar uygun eşikleme ile elde edilerek hata bölütlemesi sağlanmıştır.

Liu ve Han (2006), danışmanlı yaklaşıma benzer Gabor dalgacık filtreleri içerisinden optimal bir bireysel filtre önerdiler.

Baştürk ve ark. (2007), Gabor dalgacıklarına bağlı yeni bir yöntem önermişlerdir. Gabor dalgacıklarının filtrelemesinden elde edilen öznitelik vektörlerini azaltmak için de temel bileşen analizini kullanmışlardır. Tilda veri tabanındaki örneklem üzerinde başarılı hata tespiti gerçekleştirmişlerdir.

Liu ve ark. (2008), çalışmalarında zaman alanında ve frekans alanında öznitelik çıkarımı yapan Gabor filtreleri tasarımı önerdiler. Bu iki alandan elde ettikleri öznitelikleri parametre seçiminde kullandılar.

Mak ve Peng (2008), hatasız kumaş resminde optimal doku öz niteliklerini elde etmek için Gabor dalgacık ağını uyguladılar. Sonra, hata tespiti için iyi ayarlanmış gerçek değerli Gabor fonksiyonu uygulandı. 256×256 ebatlarında düz kumaş, kot kumaşı, kabartmalı kumaş örneklerinden alınan 39 hatasız kumaş ve 32 hatalı kumaş üzerinde, hata tespit başarı oranı %96,2 ve satır tarayıcı kamera ile alınan 259 hatasız ve 17 hatalı kabartmalı kumaş resimleri üzerinde hata tespit başarı oranı %97,1 olarak gösterilmiştir.

Han (2009), Gabor filtre maskelerine dayalı bir kumaş hata tespit yöntemi önermiştir. Bu yöntem çevrimdışı olarak, birçok hatayı içeren bir grup kumaş örnek görüntüleri üzerinde denenmiş az sayıda yanlış alarm vermiş ve doğru hata tespiti yapmıştır.

Alimohamadi ve ark. (2009), morfolojik ve Gabor dalgacık filtrelerine bağlı yeni bir hata tespit yöntemi sunmuşlardır. Gabor dalgacık filtrelerini kumaş resmine uygulayarak öznitelik matrisini elde etmişlerdir. Farklı frekans ve oryantasyonda uygulanan filtrelerin enerji yanıtına göre, optimal filtreyi seçmişlerdir. Adaptif eşikleme

için morfolojik analizi kullanmışlardır. Morfolojik analize ve optimal filtreye bağlı olarak hata tespiti gerçekleştirmişlerdir. Deneysel sonuçları başarılı ve etkilidir.

Zhang ve ark. (2010), düz kumaş hata tespitini Gaussian karışım modeli ile Gabor dönüşümünü bütünleştirerek gerçekleştirdi. 9 çeşit hata barındıran 360 hatalı resim üzerinden sınıflandırmada %87 başarı sağlamışlardır.

Chen ve ark. (2010) skala dönüşümüne bağlı Gabor filtreleme stratejisi önermişlerdir. Çalışmalarında önce hatasız kumaşa ait doku özniteliklerini elde edebilmek için, Gabor filtre bankası ile hatasız kumaş resimlerini filtrelemişlerdir. Daha sonra da aynı işlemi hatalı kumaş resimleri için uygulamışlardır. Elde edilen veriler üzerinden standart sapma, kaynaştırma ve eşikleme uygulayarak kumaş hatalarının tespitini gerçekleştirmişlerdir.

Zhang ve ark. (2010) kord bezi üzerinde hata tespitine yönelik gerçek zamanlı bir sistem tanıttılar. Gabor dalgacık kullanarak önerdikleri algoritma ile düzenli doku hatalarını tespit ettiler. Çevrimdışı ve çevrim içi denemelerinde iyi performans ve doğru sonuçlar elde etmişlerdir.

2.2.3 Dalgacık dönüşümü ile kumaş hata tespiti

Dalgacık dönüşümü çoklu çözünürlük sinyal ayrıştırma teorisi olarak Mallat tarafından sunulmuştur (Mallat, 1989). Yakın geçmişte çoklu-çözünürlük ayrıştırma (dalgacık dönüşümüne bağlı) görüntü özelliklerini çıkarmada kayda değer bir ilgi yakalamıştır. Çoklu dalgacık, bir görüntünün farklı frekanslarda konumlandırılmış alt görüntülere ayrıştırılmasına olanak sağlar. Dalgacık dönüşümü bir görüntünün 2 boyutlu frekans spektrumunda alçak geçirgen alt görüntülere ve yüksek geçirgen alt görüntülere böler. Böylece farklı frekans ve farklı çözünürlük seviyelerindeki ayrıştırılmış alt görüntülerden doku özellikleri elde edilir.

Sari-Sarraf ve Goddard (1999), resim ön işleme de dalgacık dönüşümü ve kenar füzyon kullanmışlardır. Dalgacık dönüşümü için Dauechies D2 dalgacıklarını kullanmıştır. Kumaş resimleri yüksek çözünürlüklü titreşimsiz 4096 satır tarayıcı kamera ile dokuma esnasında alınmıştır. 26 farklı tip hatayı içeren 3700 denek resmi üzerinden yaptıkları çalışmalarında %89 başarı oranını yakalamışlardır. Kim ve ark. (1999), sabit skalalar kullanmaktan ziyade kumaş hatası tespit yeteneğini arttırmak için dalgacık skalaları seçerek bir öğrenme işlemi uygulamışlardır.

Latif-Amet ve ark. (2000), dalgacık dönüşüm katasılarından ortak oluşum ve MRF (Markov Rastgele Alanları) tabanlı özellikleri elde etmişlerdir. Hataları sınıflandırmada ise dalgacık dönüşümünün alt bantlarından elde edilen katsayılar gri-seviye farkına bağlı olarak çıkarılan özellikler kullanılmıştır.

Yang ve ark. (2005) tek bir adaptif dalgacık kullanımından çoklu adaptif dalgacık kullanımına dayanan adaptif dalgacıkları tasarlamışlardır. Her bir kumaş hata sınıfı için hataya özgü adaptif dalgacık tasarlanarak dalgacık dönüşümünün bir iş parçacığında hata bölgesi belirginleştirilir. Daha sonra da hata bölgesi basit eşikleme sınıflayıcısı kullanarak tespit edilebilir. Bu çoklu adaptif dalgacık yöntemi 8 tip hata içeren 56 ve hatasız 64 kumaş görüntüsü kullanılarak %98,2 tespit oranı ve %1,5 yanlış alarm oranı ile kumaş hata tespiti sağlanmıştır.

Serdaroğlu ve ark. (2006), çalışmasında dalgacık paket dönüşümü, bağımsız bileşen analizi metotlarını birlikte kullanılarak kumaş hata tespitine yönelik farklı yaklaşım sunmuştur. Resimlere dalgacık dönüşümü uygulanarak resimlerin alt bantlara ayrılması sağlanmıştır. Çalışmalarında dalgacık dönüşüm analizi, farklı sayılarda bağımsız bileşenler ve birçok alt bant kullanılarak 13 farklı kumaş resmi üzerinde hata tespiti gerçekleştirilmiştir. Aynı zamanda uygulanan yöntemler karşılaştırılmıştır.

Liu ve Qu (2008), dalgacık dönüşümü ve geri beslemeli yapay sinir ağını birlikte kullanılarak hata tespiti ve sınıflandırılması yaptılar. Her bir hata türü için 20 adet yapay sinir ağında eğitim amaçlı ve 10 âdeti de test amaçlı olmak üzere 30 adet resim kullandılar. Bu resimler Db5 dalgacık türü ile üçüncü seviyede alt resimlere ayrılmıştır. Sonrasında ise bu görüntülerden elde edilen öznitelik vektörleri yapay sinir ağına girdi olarak verilmiştir. Yağ lekesi, kopuk çözgü ve kopuk atkı hataları dikkate alınarak gerçekleştirilen bu çalışmalarında hata sınıflandırmasında %95 oranında başarı sağlamışlardır.

Guan ve Shi (2008), dalgacık dönüşümü ve Fourier dönüşümü yöntemlerini kumaş hata tespitinde kullanmışlardır. Denim kumaşlar ve bezayağı kumaşlarını Db1, Db2, Db3, Db4 dalgacık türleri ile birinci seviyede alt resimlere ayrıştırıp, her bir alt resim için entropi değerlerini hesaplamışlardır. En düşük entropi değeri Bezayağı kumaşlarda DB2'de, denim kumaşlarda ise DB3'de elde edildiğinden bezayağı için DB2, denim kumaşlar için ise db3 dalgacık türleri seçilmiştir. Elde edilen bu alt resimlerde frekans uzayı filtresi uygulanarak gürültü temizlemesi gerçekleştirilmiştir. Filtrelenen bu alt resimler alt pencerelere bölütlenerek bu alt pencerelere ait standart sapma değerleri

hesaplanmıştır. Hatalı ve hatasız bölgeler için standart sapma değerleri arasındaki fark belirlenen limiti aşıyorsa bu bölgeler hatalı kabul edilip tespit edilmiştir. Deneysel veriler hata tespit oranın %92,5'in üzerinde olduğunu göstermiştir.

Han ve Shi (2007) dalgacık dönüşümüne bağlı kumaş hata tespiti gerçekleştirmişlerdir. Çalışmalarında, uygun seviyede ayrıştırılan yaklaşım alt resimleri zeminden temizleyerek tekrar resimler oluşturmuşlar ve yaklaşım alt resimlerin ortak oluşum matrislerinin analizine dayalı adaptif seviye seçimi şeması sunmuşlardır.

Jiang ve ark. (2009) çalışmalarında dalgacık dönüşümünün optimal ağaç yapısını önermişlerdir. Dalgacık dönüşümü ile elde edilen alt-bant resimlerinin entropi ve enerji değerlerini hesaplayarak bir kumaş hata tespiti yapmışlardır.

Liu ve ark (2010), dokunmamış kumaşları sınıflandırmada dalgacık doku analizi ve öğrenen vektör ölçekli yapay sinir ağlarını kullanmışlardır. Dalgacık analizi ile dokunmamış kumaş resimleri ayrıştırılarak detay katsayıları elde edilmiştir. YSA eğitimi ve test için giriş verileri olarak yüksek frekans bileşenleri yani detay katsayıları kullanılmıştır. Çalışmalarında beş çeşit kategori için sınıflandırma başarısı %87 olduğu görülmektedir.

Su ve ark. (2010), dalgacık dönüşümü ile dalgacık enerjilerini elde ederek bunları Taguchi tabanlı geri beslemeli yapay sinir ağına girdi parametresi olarak vermişlerdir. Taguchi yöntemi YSA'yı hızlandırma amaçlı kullanılmıştır. Çalışmalarında % 96,5 oranında bir sınıflandırma başarısı elde etmişlerdir.

Guan (2010), kumaş hata tespitine yönelik yeni bir yaklaşım sunmuştur. Çalışmalarında, Fourier dönüşümü ve dalgacık dönüşümü birlikte kullanılmıştır. Algoritmalarında Fourier filtreleme ile belirgin hatalar, tek seviyede dalgacık dönüşümü ile normal hatalar, çok seviyeli dalgacık dönüşümünü ile daha küçük hatalar tespit edilmiştir. Birinci ve ikinci seviye için Db3 dalgacık ayrıştırmasından elde edilen yaklaşım alt-resimlerinden standart sapma, entropi, enerji gibi öznitelik vektörleri elde edilmiştir. Bu öznitelik vektörleri ve geri beslemeli YSA ile 256×256 piksel 160 adet kumaş resmi üzerinden 4 farklı tip hata için %95 ile %100 arasında bir sınıflandırma başarısı sağlanmıştır.

Guan ve ark. (2011), çalışmalarında DB dalgacık ailesini kullanarak kumaş hata tespiti gerçekleştirmiştir. Ayrıştırılan alt resimlerden seçilenler tekrar oluşturulduktan sonra morfolojik işlemlerle hata tespiti yapılmıştır.

Liu ve ark (2011), çalışmalarında dokunmamış kumaşların sınıflandırmasında dalgacık enerji ve Bayesian yapay sinir ağını kullanışlardır. Kumaş karakteristiğini çıkarmak için kullandıkları dalgacık dönüşümünü 3. Seviyede gerçekleştirip 18 özellik çıkarmışlardır. Başarını oranını %98 olarak bulmuşlardır. 4. Seviyede gerçekleştirdikleri ayrıştırmadan elde ettikleri 24 özellik ile ise başarı oranını %99'a çıkarmışlardır.

Liang ve ark (2012) iplik yüzeyinin objektif ve otomatik değerlendirmesi için dalgacık dönüşümü ve istatistiksel ölçümü uygulamıştır. Ayrık dalgacık dönüşümünü uygulayarak iplik resmi üzerinden iplik tüylenmesini zeminden temizlemişlerdir. İplik yüzeyinden tüylenme karakteristiğinin elde edilmesi dalgacık enerjilerinin hesaplanması ile gerçekleştirilmiştir. İplik çap değişimi ve dağılımı, iplik hataları gibi iplik istatistiki özellikleri de kullanmışlardır. Sınıflandırma için deneysel çalışma sonuçları %90 başarı olduğunu göstermektedir.

2.3 Model Tabanlı Yaklaşım ile Kumaş Hata Tespiti

Cohen ve ark. (1991), Gaussian Markov rastgele alanları (MRF) hatasız kumaş resimlerini modellemek için kullanmışlardır. 6 adet 256×256 piksel ebadındaki çeşitli hatalara sahip kumaş resimleri 3 deney düzeneği halinde test edilmiştir. Yöntemlerinde 6 resim için tüm hataların tespit edilmesine rağmen, daha çok resim için aynı modellemenin değerlendirmesi hakkında bir bilgi bulunmamaktadır.

Basu ve Lin (1992), çoklu skala oto-regresif modelinin kullanımı üzerine çalışmışlardır. Çiçek desenli, kare desenli, halı desenli örnek kumaş resimlerini kullanmışlardır. Çalışmalarının sonucu olarak modelin hesaplama açısından hızlı ve yeterli olduğu belirtilmiştir.

Özdemir ve Erçil (1996), Markov rasgele alanlarını kumaş hata tespitinde kullanmış ve çalışmalarında MRF tabanlı yöntem ile Karhunen–Loeve (KL) tabanlı yöntemi karşılaştırmışlardır. 256×256 piksel ebatında 4 hatalı kumaş resmi hata tespiti için net sonuçlar verilmemiştir. Fakat çalışma zamanı açısından MRF 0.33 sn. iken KL 3.13 sn. olarak verilmiştir.

Chan ve ark. (2005), kumaş hata tespiti için dalgacık alan Gizli Markov Ağaç modelini sunmuşlardır. Kumaş zemin homojen doku içerdiği düşünülerek HMT ile modellenmiştir. Bu model ile Dalgacık alanında hatalı resmin bir alan haritası üretilmiştir. Bu alan haritasına bir seviye kümeli segmentasyon tekniği uygulanarak hataların

konumları elde edilmiştir. Yöntemlerini X-ray ile elde edilen hatalı kumaş resimleri üzerinde test etmişlerdir. Fakat detaylandırılmış bir değerlendirilme verilmemiştir.

2.4 Yapısal Yaklaşım ile Kumaş Hata Tespiti

Chen ve Jain (1988), kumaş hatalarını tanımlamada yapısal yaklaşım önermiştir. 128×128 piksel ebadındaki 5 adet hatalı kumaş örneği üzerinden yaptıkları çalışmalarında açık şekilde değerlendirmeler bulunmamaktadır.

Bennamoun ve Bodnarova (1998), yapısal yaklaşımı doku hata tespitinde önermiştir. Bu yaklaşımın dezavantajlarından bir tanesi de hesaplama açısından yoğun işlem gerektirmesidir. Bu yaklaşımda hata tespiti algoritması ile normalleştirilmiş çapraz korelasyonu karşılaştırılmıştır. Yaklaşımlarında başarı oranı %80 iken çapraz korelasyon da bu başarı % 95 oranındadır.

Chetverikov (2000), çalışmalarında yapısal yaklaşımı doku hata tespitinde kullanmıştır. 256×256 piksel ebadındaki 8 adet resim için iki test gerçekleştirmişlerdir. Deneysel sonuçlar hakkında açık bir değerlendirme yapılmamıştır.

2.5 Yapay Sinir Ağları ile Kumaş Hata Tespiti

Rohrmus (2000), tarafından yapay sinir ağlı kompakt bir sistem sunulmuş fakat yeterli ölçüde detaylandırılmamıştır. HME regal örme makineleri için kumaş hatalarını tespiti ve sınıflandırması yapan ANFIS'li otomatik görsel tespit sistemini önermiştir

Hung ve Chen (2001), geri yayılımlı yapay sinir ağları ve bulanık mantığı 8 farklı tip hata sınıflandırması için kullanmışlardır. Giriş verileri olarak yatay ve dikey izdüşüm uzunlukları oranı, gri seviye ortalaması, standart sapması gibi verileri kullanmışlardır. Aynı zamanda YSA ile YSA-bulanık sistemini karşılaştırmışlardır. Sonuçlarda bulanık YSA sistem sınıflandırma açısından YSA'dan daha iyi olduğu belirtilmiştir.

Kumar ve Shen (2002), YSA ve destek vektör makinesi (SVM) kullanarak doku yüzeylerinde hata tespiti için iki yöntem sunmuştur. FFN ile kumaş hata sınıflandırması hesaplama açısından yüksek işlem yoğunluğuna sahiptir. Kumar (2003), lokal kumaş hatalarının sınıflandırılması için İleri Beslemeli Ağlar ile yeni bir yaklaşım sunulmuştur. Hata tespiti için aynı zamanda düşük maliyetli çözüm olan lineer yapay sinir ağı yaklaşımı

sunulmuştur. Bu iki yaklaşımın deneysel sonuçları gerçek kumaş görüntülerinden elde edilmiştir.

Castilho ve ark. (2007), gerçek zamanlı kumaş hata tespit sistemi önermiştir. Ürün alan tabanlı bulanık modelleme, yapay sinir ağları, bulanık kümeleme ve adaptif yapay sinir ağı (ANFIS) hata tespiti için kullanıldı. Deneysel sonuçları, yapay sinir ağının hızlı bir performansa sahip olduğunu göstermektedir.

Yuen ve ark. (2009), giysi kumaş hataları sınıflandırma da genetik algoritma ve YSA'yı birleştirerek hibrid bir model sunmuştur. Örnek görüntüleri sınıflandırmak için dört karakteristik değişken, geri yayımlı sinir ağına giriş olarak verildiğinde hata tanımlamasında yüksek doğruluk oranına sahip olduğu görülmüştür.

Shi ve ark. (2009) kumaş hataları tespitinde basit birleşik darbeli sinir ağına dayalı adaptif görüntü segmentasyonunu önermiştir. Bu çalışmalarında kontrast sapması denilen yeni bir parametre tanımlamışlardır. Kontrast sapması görüntüdeki hatasız bölge ile hata arasında satır ve sütundaki kontrast farkını tanımlar. YSA parametre seçiminde yerel ve global kontrast sapma bilgisini azaltarak YSA hesaplama yükünü azaltır.

Yin ve ark. (2009), dalgacık dönüşümünün ve geri yayılımlı sinir ağının kullanıldığı diğer bir hata tespit ve sınıflandırma yöntemi sunulmuştur. Görüntü üzerinde yağ ve delik lekelerinin histogramı hesaplatılarak geri yayılımlı sinir ağına giriş parametreleri olarak verildi. Deneysel sonuçları, hataları yüksek tanıma oranına sahip bir şekilde tespit ettikleri ve sınıflandırdıklarını göstermiştir.

2.6 GPU Teknolojisiyle İlgili Önceki Çalışmalar

CUDA teknolojisinin geliştirilmesi ile paralel uygulamalarda sağlanan yüksek performans araştırmacıları görüntü işleme tekniklerinin GPU üzerinde kullanılmasına yönlendirmiştir. Yeni nesil grafik kartlarındaki çoklu çekirdek mimarisi ve programlanabilme yetenekleriyle artık karmaşık ve çok zaman gerektiren problemlerin çözümü için bu kartların etkin olarak kullanılabileceğini göstermektedir (Owens, 2008).

Garland ve ark. (2008), CUDA ile medikal alanda görüntü isleme yönelik yaptıkları çalışmada CPU ve GPU üzerindeki performans değerlendirmesi de yapmışlardır. Yang ve ark. (2008), kenar bulma, histogram alma vs. gibi görüntü islemede kullanılan bazı yöntemleri GPU üzerinde gerçekleştirmiş ve CPU ile de performans karşılaştırması yapmışlardır.

Jang ve ark. (2008), yapmış oldukları çalışmada; örüntü tanıma işlemi için kullanılan yapay sinir ağını, GPU teknolojisi üzerinde ve CPU teknolojisi üzerinde test emişlerdir. Bu çalışmaya göre 15 kat hız oranı elde edilmiştir Mak ve Tian (2010), yinelemeli gergi izleme yöntemi ile uzaysal yönlü eğim histogramı denilen çalışmalarında GPU teknolojisinden yararlanmışlardır. Bu yöntemi kumaş hata tespiti için hem CPU hem de GPU üzerinde uygulamışlardır. Bu yöntemin uygulaması GPU üzerinde CPU'ine göre 30 kat daha fazla olduğu görülmüştür.

Mattoccia ve ark. (2011), görüntü analizinde performans arttırmak amacıyla yapmış oldukları çalışmalarında GPU teknolojisi kullanmışlardır Fast Bilateral Stereo algoritması için. yapılan bu çalışmada yaklaşık 100 kat hızlandırma sağlanmıştır. Membarth ve ark. (2011), CUDA programlama modeli ile çoklu çözünürlük algoritmaları üzerine çalışmışlardır. Bellek transfer yöntemlerinde bazı farklılıklar yaparak çalışmalarında yaklaşık 145 kat hız yakalamışlarıdır.

Sonuç olarak, Literatüre bakıldığında görüntü işleme ve örüntü tanıma teknikleri ile Kumaş hataları tespitine yönelik yapılan çalışmaların başarılı sonuçlar verdiği görülmektedir. Ancak bu çalışmalarda kullanılan yöntemlerin hesap işlem yoğunluğu gerektirdiğinden gerçek zamanlı uygulama için uygulanabilirliği oldukça güçtür. GPU teknolojisinin hızlı bir şekilde gelişmesiyle birlikte bu teknolojiyi kullanarak görüntü işleme ve örüntü tanıma yöntemi ile ilgili birçok alanda yapılan çalışmaların arttığı görülmektedir. Ancak kumaş hata tespitine yönelik GPU ile gerçekleştirilmiş sınırlı sayıda çalışma olduğu görülmektedir.

Literatürden yararlanarak elde edilen veriler kumaş hataları tespiti ve sınıflandırmasında YSA ve dalgacık dönüşümünün diğer hata tespit ve sınıflandırma yöntemlerine göre daha başarılı sonuçlar verdiğini göstermektedir. Ancak bu yöntemler, bilgisayar ortamında yoğun hesaplama zamanı gerektirdiğinden çevrim-içi (online) kumaş hata tespiti ve sınıflandırma için hızlı yanıt verememektedir. Bu nedenle yeni nesil GPU ve CUDA teknolojisi kullanımının işlem hızı açısından yüksek performans sağlayacağı öngörülmektedir.

3. MATERYAL ve METOD

3.1 MATERYAL

Kumaş hatası tespiti ve sınıflandırma için bu çalışmada aşağıdaki donanımlar ve yazılımlar kullanılmıştır.

Kullanılan Donanımlar:

- Bilgisayar
- Dijital Kamera
- CUDA Destekli Ekran Kartı

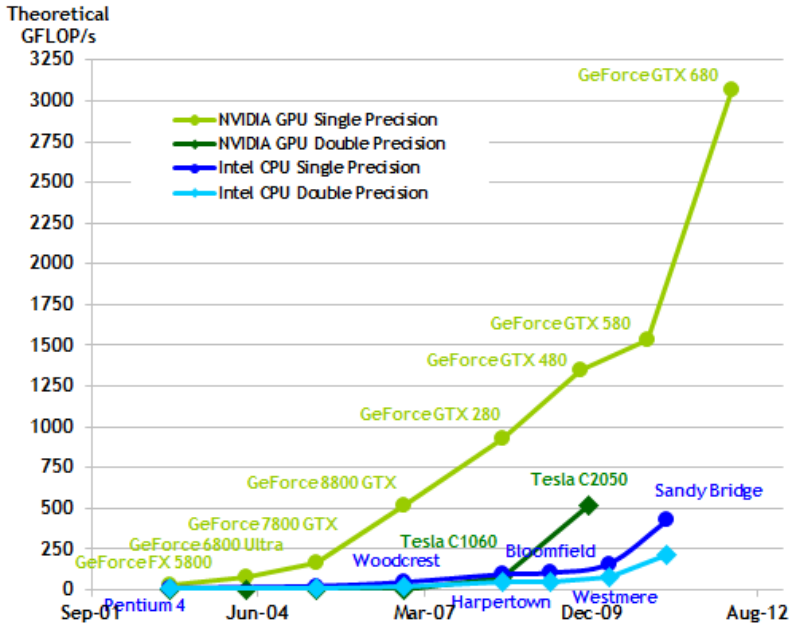
Kullanılan Yazılımlar:

- CUDA Toolkit
- MATLAB
- Jacket

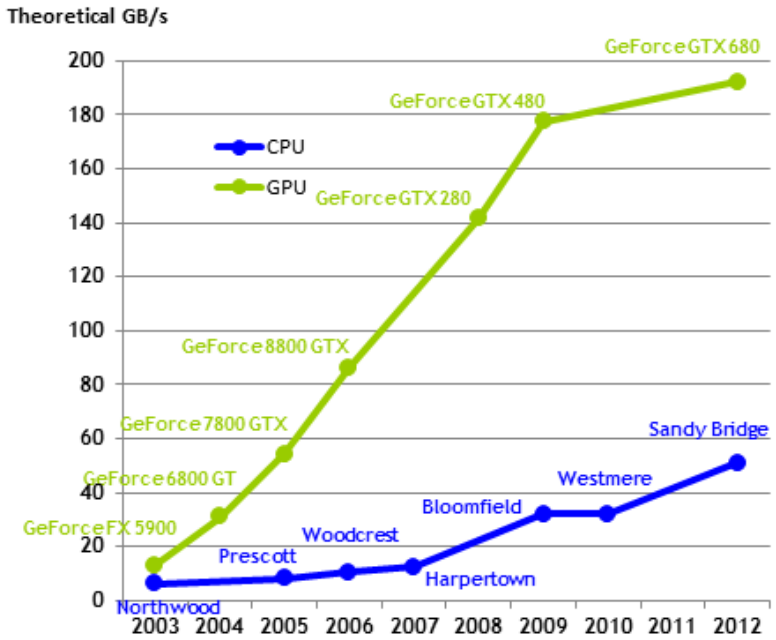
3.1.1 GPU ve CUDA

Ekran kartları temel olarak grafik derleme, işleme sorunlarına çözüm getirmek için geliştirilmiştir. Grafik işlem birimleri (GPU) ilk olarak iki boyutlu grafikleri işlemek amacıyla geliştirilmiş donanımlar olarak ortaya çıktı. Daha sonra grafik kartı işlemcilerindeki teknolojik gelişmeler 3 boyutlu grafik dönüştürmeyi, işlemeyi, piksel gölgelemeyi olanaklı hale getirdi. Nvidia'nın Geforce 3 model serisi ile de GPU genel amaçlı işlemler için kullanılmaya başlandı. Verilerin saklandığı dizi ya da matrisler doku formuna dönüştürülerek bu veriler üzerinde grafiksel işlemler yapılıyormuş gibi genel işlemler yapılabilmekteydi. Nvidia'nın Cg adını verdiği C benzeri basit bir programlama dili ile kodlanabilmekte ve programcıyı genel amaçlı işlemlerin, karmaşık gölgelendirme işlemleriyle çözümlenebilmesinden kurtarmaktaydı. Kasım 2006 da, NVIDIA tarafından genel amaçlı paralel hesaplama mimarisini CUDA (Compute Unified Device Architecture-Tümleşik Hesap Aygıtı Mimarisi) tanıtıldı. Bu yapı yeni bir paralel programlama modeli ve komut seti yapısını içermekteydi. Aynı zamanda, NVIDIA'daki paralel hesap motoru GPU'lar çoğu karmaşık hesap uygulamalarını dört çekirdekli CPU'dan daha verimli ve performanslı bir şekilde çalıştırılmasını sağlamaktadır. 2000'lı yıllardan bu yana GPU teknolojisindeki gelişmeler hesaplama işlemleri için yeni bir yön tayin etmiştir. Yıllara

göre GPU ve CPU için saniyedeki kayan noktalı işlemler Şekil 3.1'de, bellek bant genişlikleri Şekil 3.2'de görülmektedir. (NVIDIA, 2012)

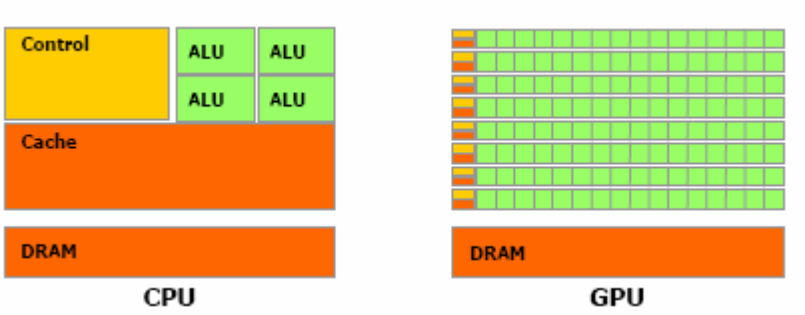


Şekil 3.1. GPU ve CPU için saniyedeki kayan noktalı işlemler



Şekil 3.2. GPU ve CPU için bant genişlikleri

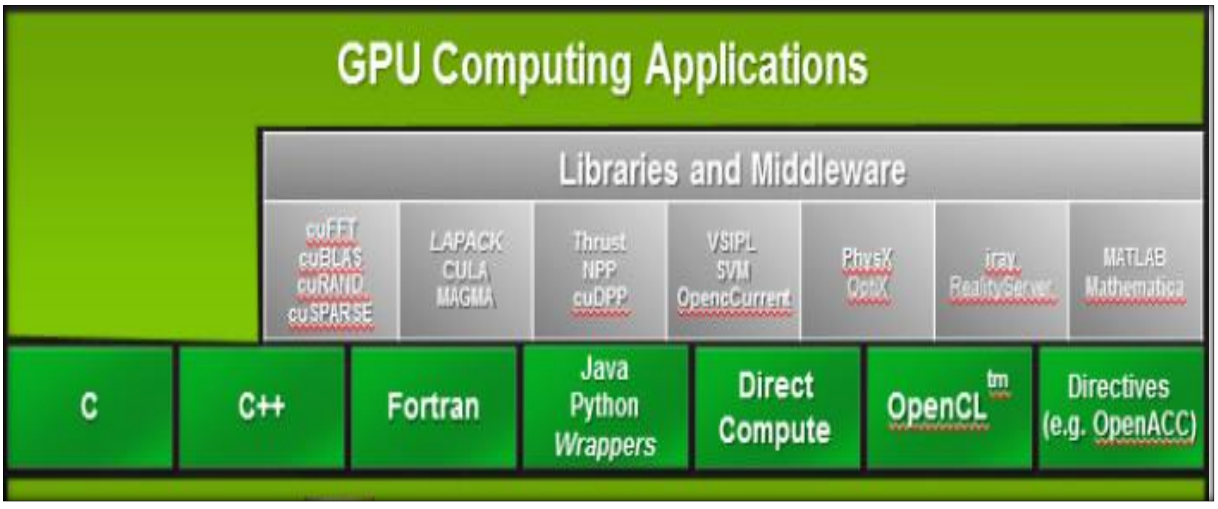
CPU ve GPU'nun temel yapısı Şekil 3.3'deki gibidir (NVIDIA, 2012). GPU şekilde de görüldüğü üzere veri işlemek üzere tahsis edilmiş daha fazla transistöre sahiptir. Bu da yüksek derece paralel hesap yoğunluğuna sahip işlemeleri CPU'ya göre çok daha hızlı bir şekilde gerçekleştirebilir.



Şekil 3.3. GPU ve CPU yapıları

CUDA, Nvidia şirketi tarafından geliştirilen GPU'lar için özelleştirilmiş grafik kodlarına ihtiyaç duymadan genel amaçlı programlama imkânı sağlayan bir yazılım geliştirme ortamıdır. Bu yazılım ortamı geliştiricilere C gibi yüksek seviye bir programa dilini kullanmayı da beraberinde getirmektedir. Bunun anlamı eski Genel Amaçlı Grafik İşlem Birimi Programlamasından (GAGİBP- GPGİB) farklı olarak üzerinde çalışılan ortamın bir GPU olduğunu kullanıcıdan soyutlamaktadır. Paralel uygulamalar geliştirirken grafik programlamaya dair bilgiye sahip olmaya yâda veri matrisini GPU üzerinde işlenecek hale getirmek için ayrıca kod yazmaya gerek yoktur. GPU'yu kodlamayı basit ve uyumlu hale getirecek, temel ve minimal anahtar kelimeler C diline eklenmiştir. CUDA kodu GNU Derleyici Koleksiyonu, Microsoft Visual Studio derleyicisi (cl) veya İntel C++ Derleyicisi(icc) kullanabilen NVIDIA C derleyicisi (nvcc) tarafından derlenir ve çalıştırılır. (NVIDIA, 2012)

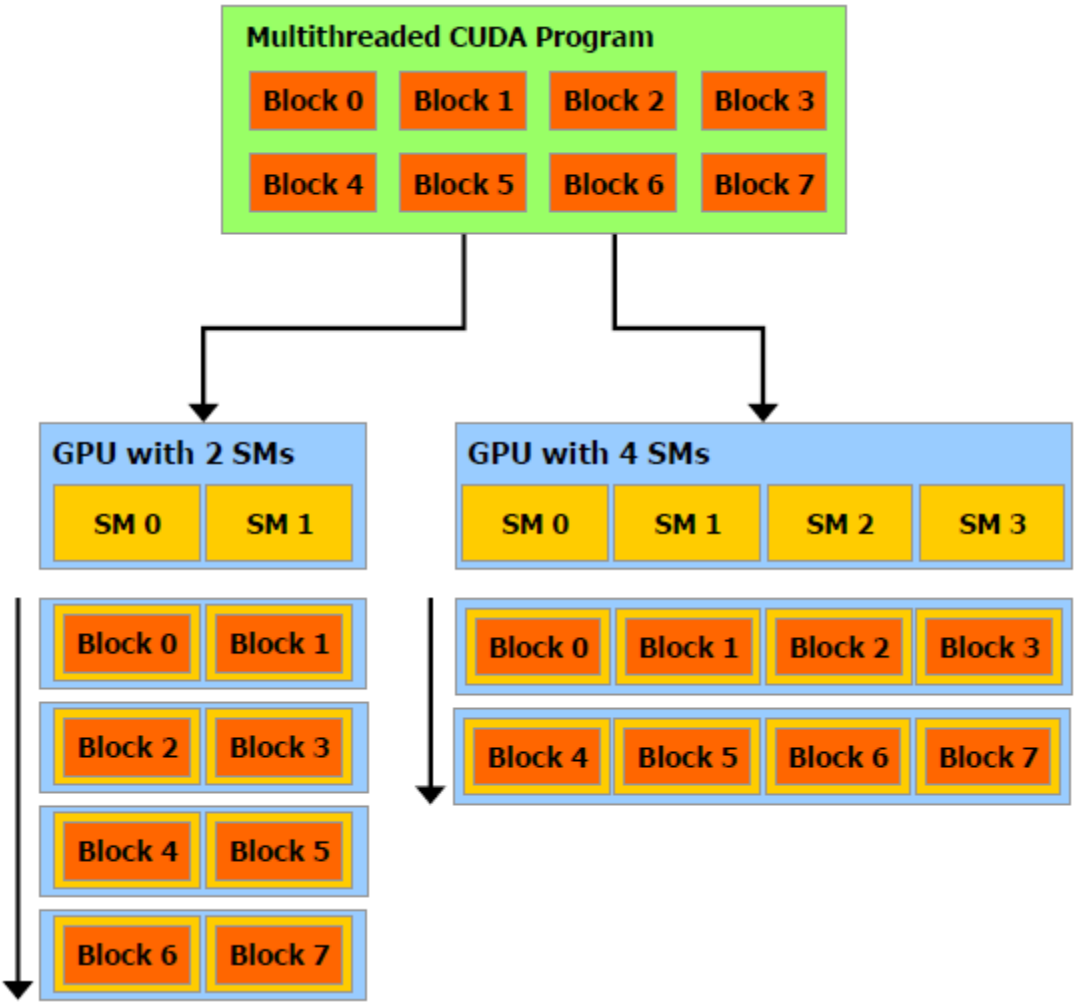
Şekil 3.4'de görüldüğü gibi CUDA çeşitli programlama dilleri ve uygulama geliştirme ve programlama ara yüzlerini desteklemedir. (NVIDIA, 2012)



Şekil 3.4. CUDA'nın desteklediği platformlar

CUDA paralel programlama modeli, standart C programlama dili gibi dillere aşina programcılara kodlamayı daha kolay yapabilmeleri için tasarlanmıştır.

GPU'nun çekirdeğinde 3 temel kavram vardır; İş parçacıkları Kümesi, Paylaşımlı Bellekler ve Bariyer Eşleme. Bu kavramlar programcının minimal bir dil eklentisi ile programlamasını kolaylaştıran kavramlardır. Aynı zamanda veri ve iş parçacığı paralelliğini sağlamaktadır. Ek olarak, problemi iş parçacığı blokları ile birbirinden bağımsızca paralel olarak çözülebilen alt problemlere bölerek çalıştırılmasına olanak sağlar. Otomatik ölçeklenebilirliği olanaklı kılarak, her bir iş parçacığı bloku GPU'inde uygun (müsait) çoklu işlemciler üzerinde çalıştırılmak üzere herhangi bir sırada, eş zamanlı yâda, sırasal olarak planlanabilir. Derlenmiş bir CUDA programı herhangi bir sayıda çoklu işlemciler üzerinde çalıştırılabilir (Şekil 3.5). Fakat Çalışma zamanı (runtime) sistemi fiziksel çoklu işlemcilerin sayısını bilmeye ihtiyaç duyar. (NVIDIA, 2012)



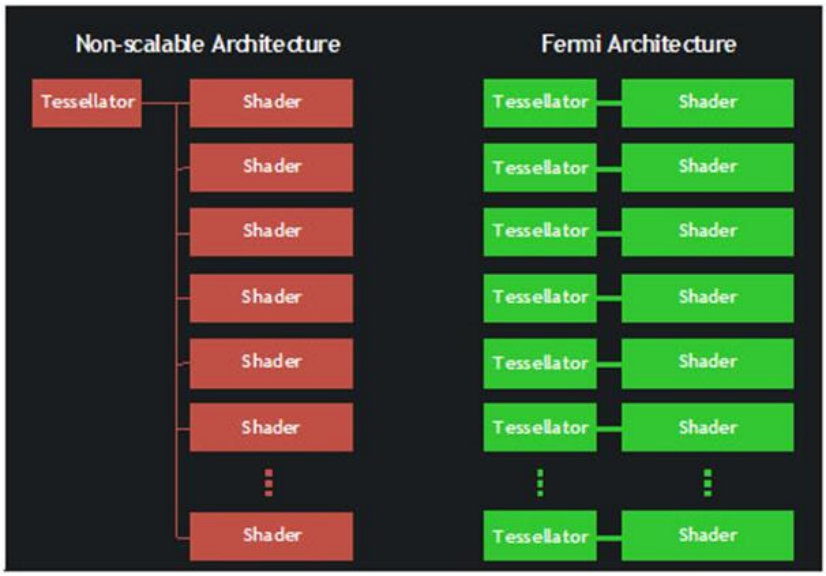
Şekil 3.5. Çoklu iş parçacıklarına sahip CUDA programı

3.1.2 Fermi mimarisi

Fermi mimarisi ile daha önceki ürünlerden elde edilen kazanımlara bağlı olarak, yeni nesil oyunlar ve uygulamalar için optimize edilmiş tamamıyla yepyeni bir mimari geliştirilmiştir.

3.1.3 Paralel Tesselasyon Motorları

Geleneksel GPU Tesselasyon işlemlerini gerçekleştirmek için tek bir geometri kullanır. Bu da piksel gölgelemeyi yapmak için tek bir piksel iletişim yolu kullanan önceki GPU tasarımına benzeşen bir tasarımdır. Fakat Fermi GPU'leri 16 paralel tesselasyon birimine kadar bu işlemi gerçekleştirebilir ve her bir birim kendi ayrılmış gölgeleme kaynaklarına sahiptir. Bu özellikler tesselasyon, rasterleştirme ve gölgeleme için devasa ölçüde bant genişliği ve yüksek çalışma verimliliği sağlar. Kıyaslandığında, Fermi GPU'leri Microsoft DirectX 11 yazılım geliştirme kitine göre 8 kata kadar daha hızlıdır. Fermi Mimarisinin gösterimi Şekil 3.6'daki gibidir.

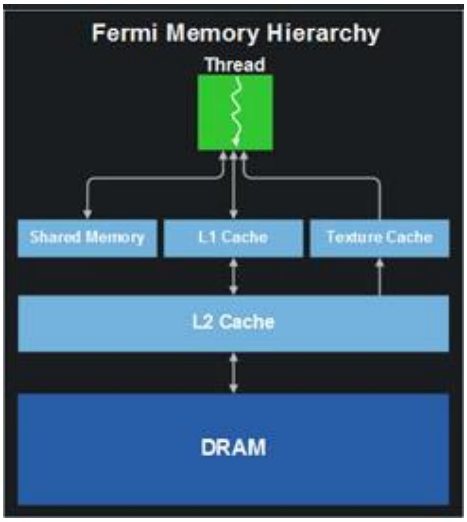


Şekil 3.6. Fermi mimarisi

3.1.4 Yeni önbellek mimarisi

Fermi mimarisi tümüyle önbellek erişimine sahip ilk GPU mimarisidir. Önbellekler uzun zamandır CPU'lerinin performanslarını arttırmak amacıyla kullanılmaktadır. Fermi mimarisi, grafikler ve programlar için önbellek sağlayan birleşik önbellek mimarisine sahiptir. Fermi programları bir doku önbelleği, bir L1 ve bir L2 önbellek erişimine sahiptir. L1 ve L2 önbellekleri, rastgele bellek erişimi isteyen üç boyutlu görüntü oluşturma gibi işler için performans artırımı sağlarlar. Doku önbelleği ise doku filtrelemesi işleminde hız

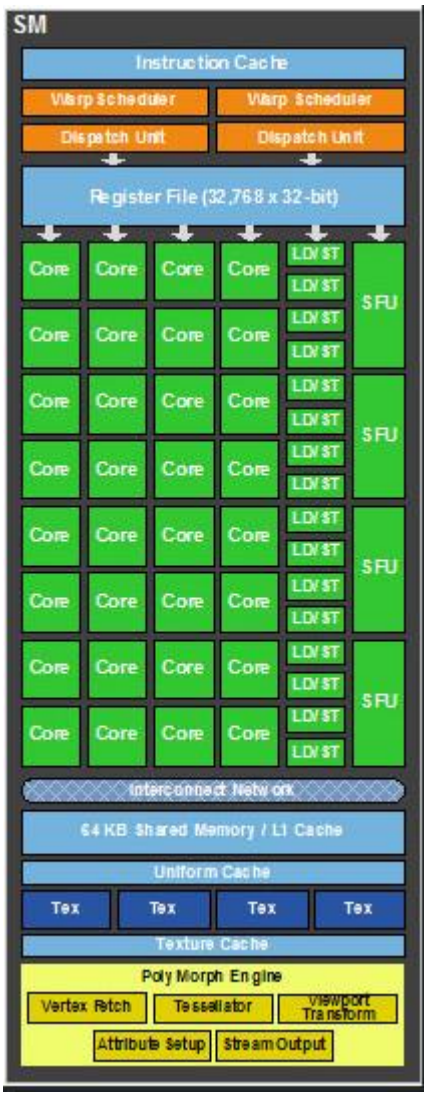
ve verimi yükseltir. Bunlara ek olarak, görüntü işleme ve video dönüştürme gibi GAGİB (GPGİB) uygulamaları için hızlı, ayrılmış ve paylaşılan bir bellek erişimine sahiptir. (Şekil 3.7) (NVIDIA, 2012)



Şekil 3.7. Fermi bellek hiyerarşisi

3.1.5 Üçüncü nesil akım (streaming) çoklu işlemcileri

Her bir Fermi AÇİ (Akım Çoklu İşlemcileri), önceki AÇİ'lere göre dört kat daha fazla olan 32 CUDA işlemcisi içerir. CUDA çekirdekleri dokular, gölge haritaları ve karmaşık gölgelendiriciler gibi çeşitli iş yüklerini tam performans ile gerçekleştirir. Her bir CUDA işlemcisi tam bir iletişim yoluna sahip tam sayı aritmetik mantık birimi (ALU-arithmetic logic unit) ve kayan nokta birimi (FPU-floating point unit) barındırır. AÇİ warp denilen 32 şerli grup halindeki iş parçacıklarının çalışmasını planlar. Her bir AÇİ iki warp planlayıcısı ve yönerge gönderi birimlerine sahiptir. Bu yapı minimal donanım gereksinimleri ile yüksek performans sağlar. (Şekil 3.8) (NVIDIA, 2012)



Şekil 3.8. Akım çoklu işlemcileri yapısı

Geliştirilen uygulama için kullanılan bilgisayarın teknik özellikleri Tablo 3.1'de ve bu bilgisayarda kullanılan NVIDIA'nın Fermi mimarisine sahip GeForce GTX 480 ekran kartı teknik özellikleri Tablo 3.2'de görülmektedir.

Tablo 3.1. Kullanılan bilgisayarın teknik özellikleri

CPU	Intel Core2 Quad CPU Q8300 4M Cache, 2.50 GHz, 1333 MHz FSB 4 Çekirdek 4 İş parçacığı
Bellek	3 GB Band Hızı 12.8 GB/s
Ekran Kartı	Nvidia GTX480 L2: 16 KB > 48 KB L1: 48 KB > 16 KB
Harddisk	320 GB 7500 rpm

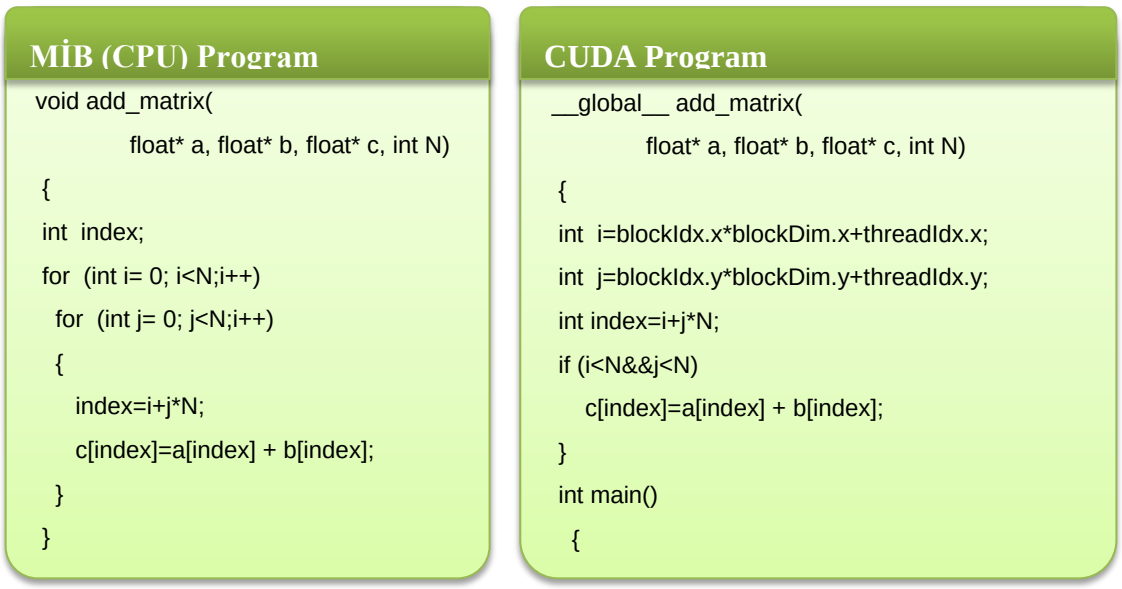
Tablo 3.2 NVIDIA GTX 480 teknik özellikleri

Adı	GeForce GTX 480
Toplam Global Hafıza	1610612736
Her Bloktaki Shared Bellek	49152
Her Bloktaki Kaydedici Sayısı	32768
Warp Boyutu	32
Bellek Alanı	2147483647
Her Bloktaki Maksimum Thread Sayısı	1024
Maksimum Thread Ölçüleri	1024,1024,64
Maksimum Grid Ölçüleri	65535,65535,65535
Clock Rate	1401000
Toplam Constant Bellek	65536
Majör	2
Minör	0
Texture Gruplama	512
Multi İşlemci Sayısı	15
Bellek Boyutu	1,5 GB DDR5
Bellek Ban Hızı	134.490 GB/s
L2 Cache	16 KB > 48 KB
L1 Cache	48 KB > 16 KB

3.1.6 Cuda programlama

CUDA CPU'nun host olarak GPU'nun da device olarak davrandığı heterojen bir hesaplama yapısındadır. CUDA C, standart C dilini eklentiler yaparak ve kullanıcılara kernel denilen C fonksiyonlarını tanımlama imkânı sağlar. Kernel fonksiyonları, normal C fonksiyonlarından farklı olarak N kere çalıştırıldıklarında, N tane ayrı iş parçacığında paralel olarak çalışırlar. Kernel fonksiyonları __global__ ifadesi ile tanımlanır. Kerneli çalıştıracak iş parçacığının sayısı <<<...>>> ifadesi ile belirtilir. Kerneli çalıştıran her bir iş parçacığına özel bir anahtar değer (ID) atanır. Bu değere "threadIDx" değişkeni üzerinden ulaşılır.

CPU ve CUDA programlarına örnek aşağıda verilmiştir. (Şekil 3.9) İlk parametre çalıştırılacak blokların sayısı, ikinci parametre is bloklarda çalıştırılacak iş parçacıklarının sayısıdır (NVIDIA, 2012).



Şekil 3.9. CPU programı ve CUDA programı

Yukarıdaki __ global__ niteleyici bir fonksiyonu kernel tanımlar. Fonksiyon GPU üzerinde çalışır ve sadece host'tan çağrılabilir. __global__ fonksiyonları void döndürme tipine sahip olmalıdır. __global__ fonksiyonu çağıldığında eş zamanlı olmadığından GPU (device) den daha önce değer –sonuç- döndürür.

__device__ niteleyici bir fonksiyonu tanımlar, bu fonksiyon GPU üzerinde çalışır ve sadece GPU üzerinden çağrılır.

__host__ niteleyici bir fonksiyonu tanımlar, bu fonksiyon host(PC) üzerinde çalışır ve sadece host üzerinden çağrılır.

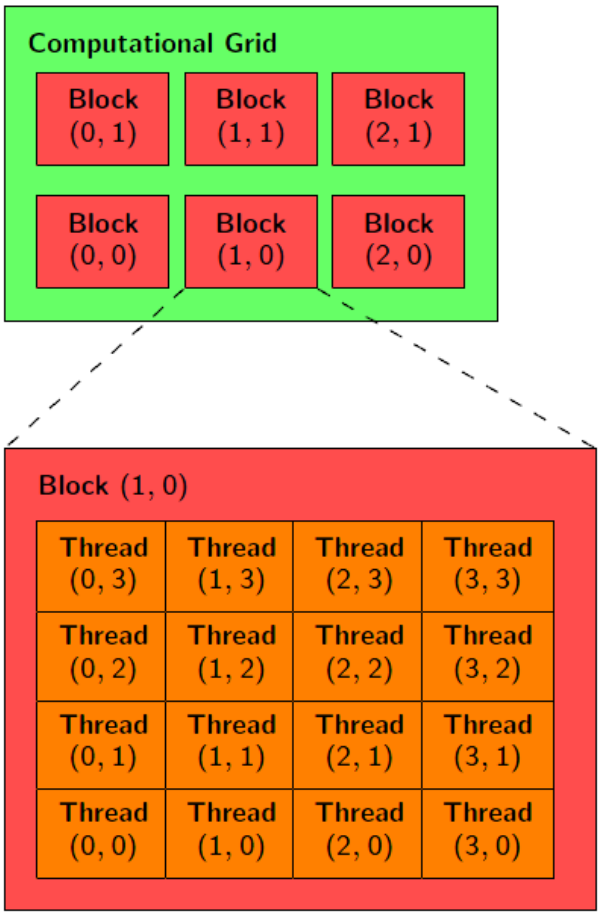
gridDim dim3(integer vektör tipi olup boyutları belirler) tipinde bir değişkendir ve ızgaranın (grid) boyutlarını içerir.

blockIdx uint3 tipinde bir değişkendir ve ızgaradaki blok indeksini içerir.

blockDim dim3 tipinde bir değişkendir ve blok boyutlarını içerir.

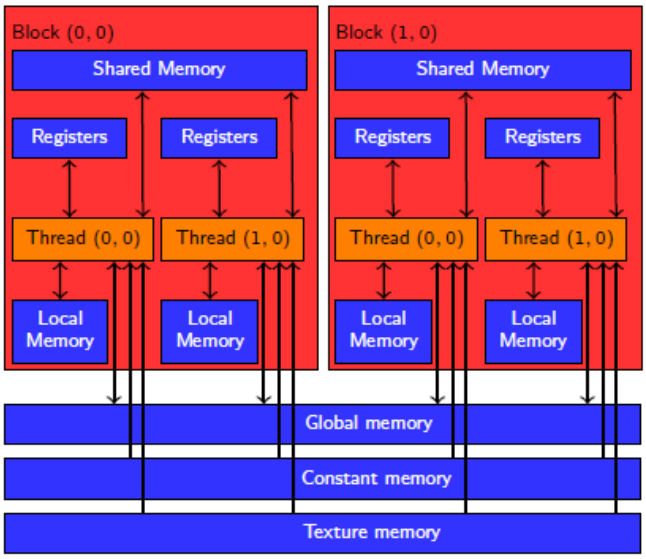
threadIdx uint3 tipinde bir değişkendir ve blok içerisindeki iş parçacığının indeksini içerir.

Izgara, blok ve iş parçacığının temel yapısı Şekil 3.10'da verilmiştir. Bir kernel için bir tek ızgara varken, bir ızgara içerisinde 3 boyutlu belirli sayıya kadar bloklar vardır ve bu bloklar içerisinde de paralel çalışabilen iş parçacıkları vardır.



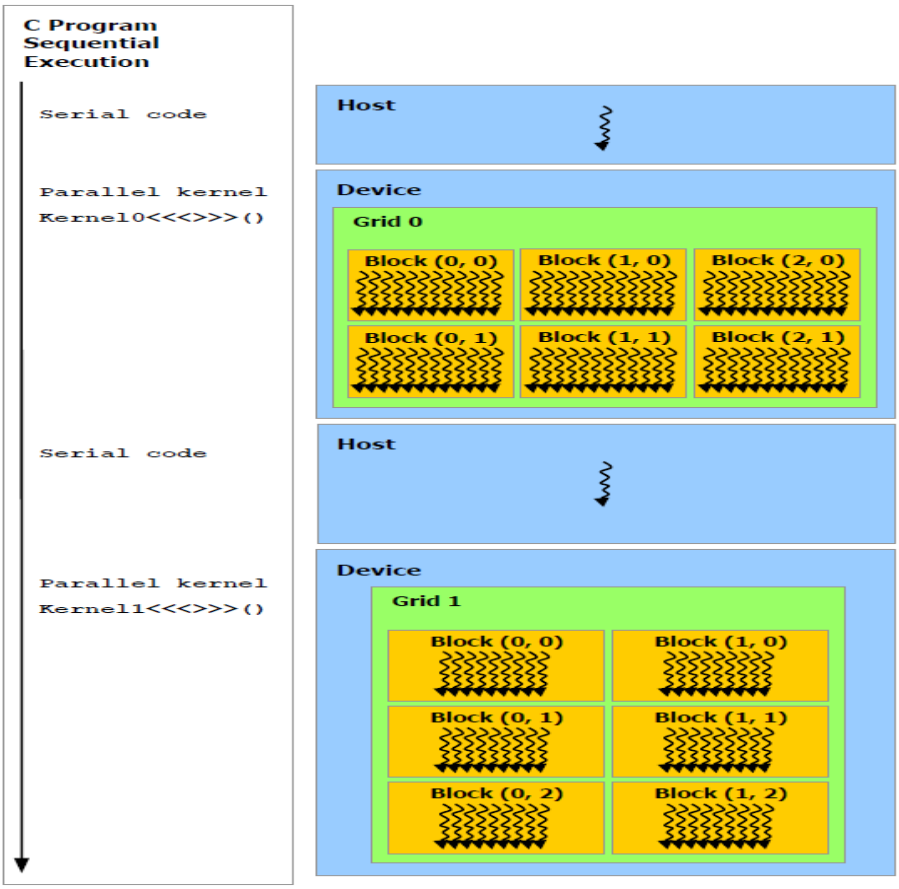
Şekil 3.10 Izgara, blok ve iş parçacığının temel yapısı

Blok ve iş parçacıklarını içeren bu ızgara yapısı CUDA sayesinde paralel işlemeyi olanaklı hale getirir. CUDA iş parçacıklarının birçok bellek türüne erişimi vardır. Şekil 3.11'de iş parçacıkları ve erişebildikleri bellek türleri gösterilmektedir. Her bir iş parçacığı özel erişimi bulunan bellek alanına sahiptir. Her bir blok ise içerisindeki bütün iş parçacıkların erişimine açık olan ortak bellek alanına sahiptir. Yani bir blok içerisindeki iş parçacıkları birbirleriyle iletişimi paylaşılan bellek üzerinden gerçekleştirirler. Bu bellek türünün ömrü, ilgili bloğun ömrü ile sınırlıdır. Bütün iş parçacıkları aynı global belleğe erişim hakkına sahiptirler. Ama bir iş parçacığı farklı bir bloktaki iş parçacığı ile paylaşılan bellekle iletişim kuramazken, yalnızca global bellek ile iletişim kurabilirler. Paylaşılan belleğe erişmekle çekirdek kayıtlarına erişme hızı aynıdır. Fakat, bu durum global bellek için aynı değildir. Bu sebeple, programcı aynı işi yapabilmeleri için iş parçacıklarını ayarlamalıdır.



Şekil 3.11 İş parçacıkları bellek erişim türleri

CUDA programlama modeli çalıştırılan uygulamanın C programlama dili ile yazılmış sıralı düzendeki kısımlarını CPU, geriye kalan ve GPU üzerinde çalışan paralel kısımları ise CUDA yönetir. Şekil 3.12'te örnek bir GPU uygulaması için program akışı görülmektedir.

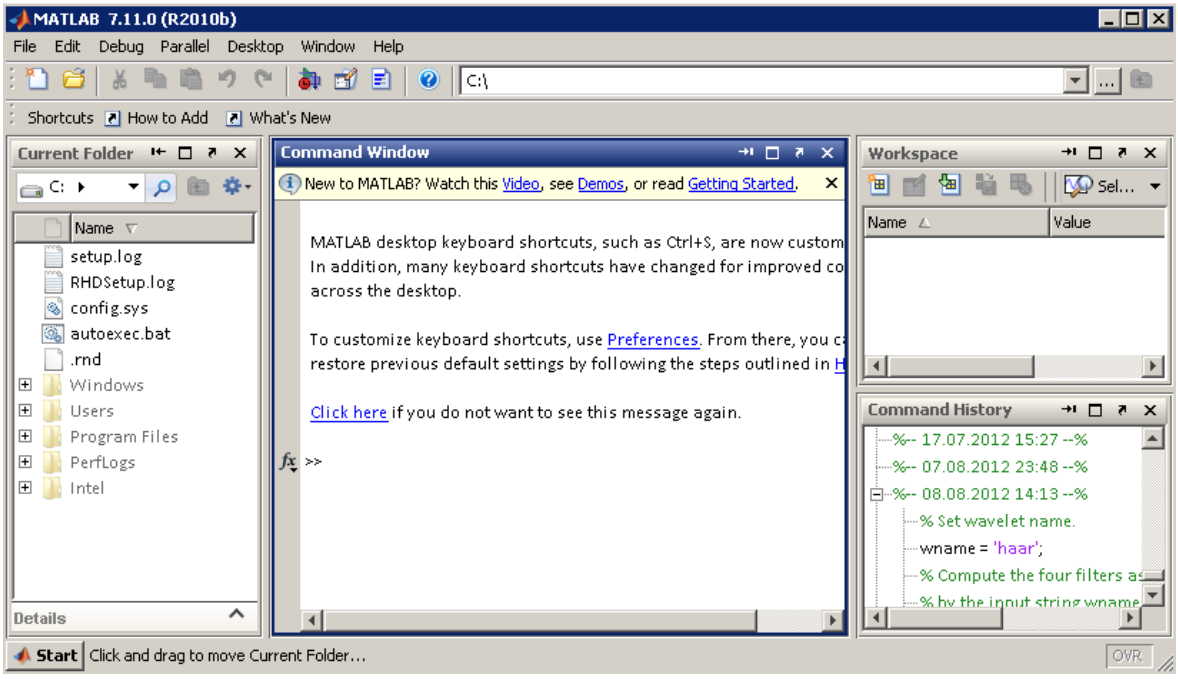


Şekil 3.12 CUDA C için örnek program akışı

CUDA programlama modeli CPU ve GPU'lerinin kendilerine ait ayrı bellek alanları olduğunu varsayar (host bellek, device bellek). Bu nedenle programlar global, sabit ve doku(texture) bellek alanlarını CUDA çalışma zamanını (runtime) kullanarak yönetirler. Bu yönetim ayrıca CPU ve GPU (host ve device) arasındaki veri iletişimini yönettiği gibi, GPU (device) bellek tahsisi, bu tahsisin kaldırılması gibi işleri de yönetirler. (NVIDIA, 2012)

3.1.7 MATLAB

Adını 'Matrix Laboratory' kelimelerinden alan MATLAB, sayısal ve sembolik hesaplamalar, veri çözümlemesi, gerçek ortamda test ve ölçüm yapabilme, çok gelişmiş çizim işlemleri, algoritma geliştirme, ileri seviye programlama, C/C++ ile tümleşik çalışabilmesi gibi olanakları sağlayan bir yüksek seviye programlama dili ve aynı zaman etkileşimli bir çalışma -yazılım- ortamıdır. MATLAB sağladığı kullanım kolaylığı, esnekliği ve fonksiyonel bir yapıya sahip olmasından kaynaklı tüm dünyada birçok alanda yaygın olarak kullanılmaktadır. MATLAB Şekil 3.13'teki gibi ekran görüntüsüne sahiptir.



Şekil 3.13 MATLAB programı arayüzü

MATLAB çalıştırıldığında şekildeki gibi bir arayüz karşımıza gelir. Bu arayüzde aşağıdaki paneller bulunmaktadır:

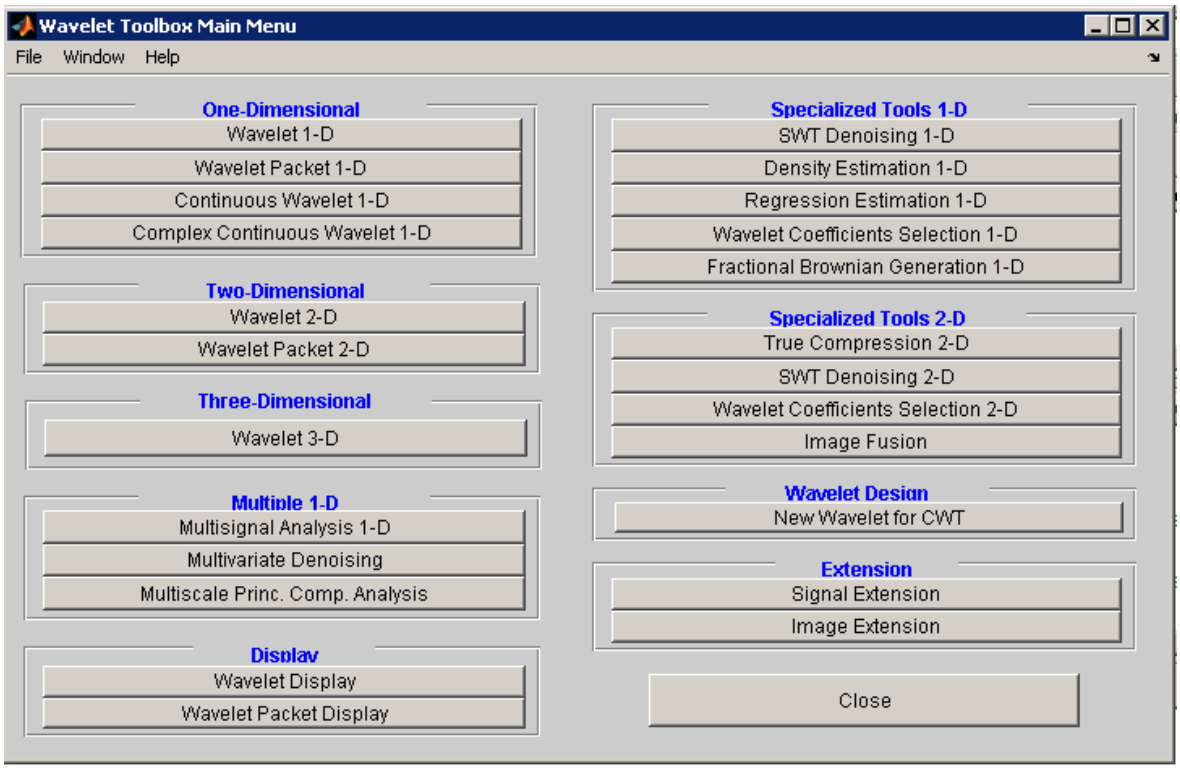
Current Folder (En Son Çalışılan Klasör);dosyalara erişimi arayüz üzerinden sağlar.

Command Window (Komut Penceresi);çalıştırılacak komutların girildiği alandır.

Workspace (çalışma alanı);kullanıcı tarafından oluşturulmuş ve dışarıdan eklenmiş verilerde izleme sağlar.

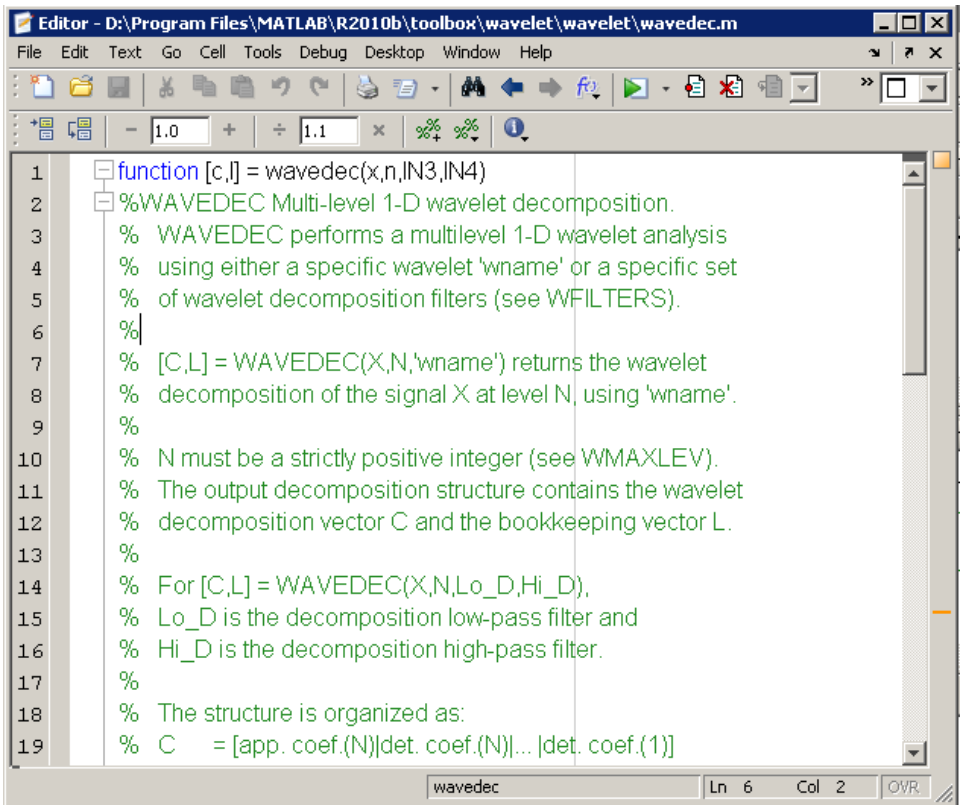
Command History (komut geçmişi);çalıştırılmış komutların görüntülenesini veya tekrar çalıştırılmasına olanak veren alandır.

MATLAB'da kullanıcı komut penceresinde komutlarla değişken oluşturma, fonksiyon çağırma gibi birçok işlemi gerçekleştirebilir. Ayrıca içerisinde bulunan birçok araç kutusuna komut penceresinden erişilebilir, örneğin komut penceresinden >>*wavemenu* komutu çalıştırıldığında dalgacık dönüşümünde kullanılan işlevleri içeren bir arayüz olarak karşımıza gelecektir.(Şekil 3.14)



Şekil 3.14. Dalgacık araç kutusu

Bu arayüz, kullanıcı tarafından etkileşimli olarak kullanılabileceği gibi araç kutusu tarafından sağlanan hazır fonksiyonlarla aynı işlevleri komut penceresi üzerinden veya fonksiyon oluşturarak gerçekleştirebilir. Şekil 3.15 fonksiyon oluşturma penceresinin görüntüsünü içermektedir.



Şekil 3.15 Dalgacık araç kutusuna ait wavedec hazır fonksiyonu

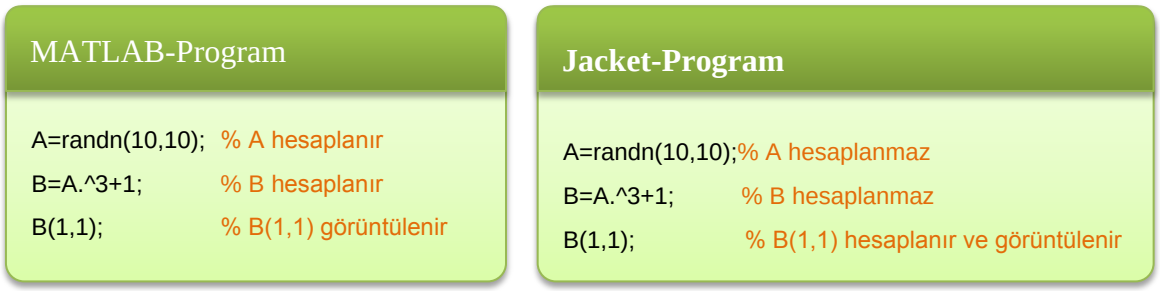
3.1.8 Jacket

NVIDIA 2006 yılında geliştirmiş olduğu CUDA teknolojisi ile GPU programlamaya yönelik bir takım düşük seviye programlama araçları ortaya çıkmıştır. Jacket ara yazılımı kullanıcın düşük seviye dilinin karmaşıklığından kurtarmak amacıyla geliştirilmiştir. Jacket yüksek seviye dili olan M programlama dili (Matlab'a ait) ile GPU'yu bir birine bağlar. Matlab programı için Jacket, GPU ile arka tarafta çalışan bir hesaplama motoru gibidir ve önemli üç özelliği beraberinde sunar; hesaplama hızı, görsellik ve kullanıcı dostluğu. Jacket otomatik olarak M programlama dili kodlarını paketleyerek GPU ile uyumlu hale getirir. Basitçe giriş verilerini kendi veri yapısına dönüştürerek, M kodunun GPU üzerinde çalışmasını sağlar. Jacket, GPU derleyicisine gerçek zamanlı ve şeffaf erişim sağlayarak, M dilinin yorumlayıcı doğasını korur.

Performans amaçlı Matlab'da M-kodunu optimize etmek, genelde C/C++ programlama dilinde MEX arayüzünü kullanarak fonksiyonların tekrar yazılmasıyla sağlanır. Çoklu işlemci ve çoklu CPU sistemleri benimsendiği için, C/C++ kullanıcıları MPI (Message Passing Interface) veya OpenMP (Open Multiprocessing) yönelirken, Matlab kullanıcıları da PCT (Paralel Computing Toolbox) ve MDCS'ye (MATLAB Distributed Computing Server) yönelmiştir. Birçok geliştirilmiş paralel araçlarla bile,

paralelleştirme işlemi zor olmakla birlikte ve zaman kaybına neden olmaktadır. Diğer taraftan da GPU üzerinde veri paralelleştirmesi es geçilerek tamamen CPU üzerine yoğunlaşılmıştır.

Matlab PCT ile Jacket kıyaslandığında GPU üzerinde çalışabilen daha fazla fonksiyon sahiptir ve fonksiyonlar daha hızlıdır. Diğer taraftan iyi tasarlanmış ve kodlanmış CUDA kodu Jacket'den %20 kadar hızlıdır.



Şekil 3.16 MATLAB ve Jacket algoritması

Jacket, komutların toplandığı tembel hesaplama şeması kullanır ve paralellik için bunu optimize etmeye çalışır (Şekil 3.16). Yazılı koda bakıldığında anlaşıldığı üzere Jacket sonuç istendiği zaman hesaplamayı gerçekleştirir.

Bu tez çalışmasında geliştirilen uygulama MATLAB-M programlama dilinde CPU'da ve GPU'da çalıştırılmak üzere tasarlanmıştır. GPU'da MATLAB-M dilinin çalışması için ise Jacket programı kullanılmıştır. Düzgün ve sağlıklı ölçümler yapabilmek geliştirilen uygulamada Jacket ve MATLAB programının desteklediği hazır fonksiyonlar ve kodlar kullanılmıştır (Jacket, 2012).

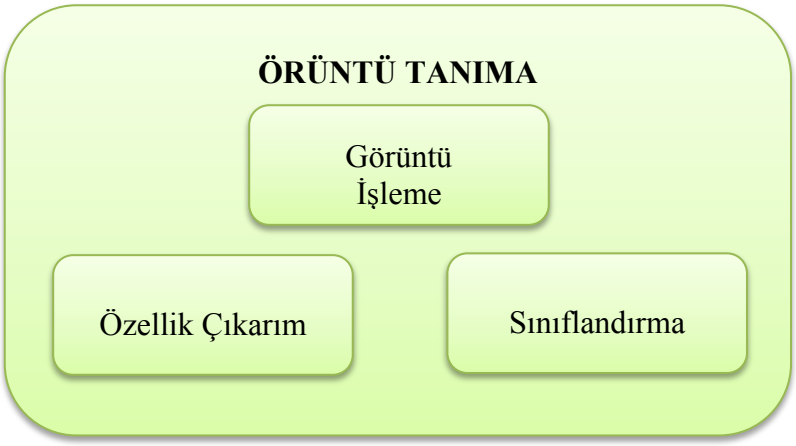
3.2 METOD

Bu tez çalışmasında, otomatik olarak kumaş hata tespit ve sınıflandırması için bilgisayarla görme kontrollü bir uygulama geliştirilmiştir. Bu uygulama, örüntü tanıma algoritmalarını içermektedir. Bu algoritmalar MATLAB+CPU ve Jacket yardımıyla MATLAB+GPU şeklinde çalıştırılmıştır. Sistem performansının yani CPU ve GPU arasındaki işlem hızının sağlık ölçülebilmesi için M programa dili ile GPU arasında ara bir yazılım olarak çalışan Jacket programının desteklediği fonksiyonlar ile MATLAB programının desteklediği ortak fonksiyonlar temel alınarak uygulama geliştirilmiştir. (Örneğin, MATLAB programı dalgacık dönüşümünde ayrıştırma için kullanılan wavedec komutunu desteklerken Jacket bu komutu desteklememektedir.). M programlama dilinde

geliştirilen uygulama da MATLAB programı ve Jacket programı için aynı algoritma ve kodlar kullanılarak yazılmıştır.

Örüntü tanıma işlemi aralarında ortak özellik bulunan ve bir ilişki kurulabilen karmaşık işaret örneklerini veya nesneleri bazı tespit edilmiş özellikler veya karakterler vasıtası ile tanımlama veya sınıflandırma işlemidir. Bu bağlamda, örüntü tanımanın en önemli amaçları; bilinmeyen örüntü sınıflarına belirli bir şekil vermek ve bilinen bir sınıfa ait olan örüntüyü teşhis etmektir. (Türkoğlu, 2002)

Örüntü tanıma teknikleri birçok mühendislik, tıp, askeri ve bilim alanında kullanılmaktadır. Örüntü tanıma işlemi, Şekil 3.17'de gösterildiği gibi üç önemli birimden oluşmaktadır:



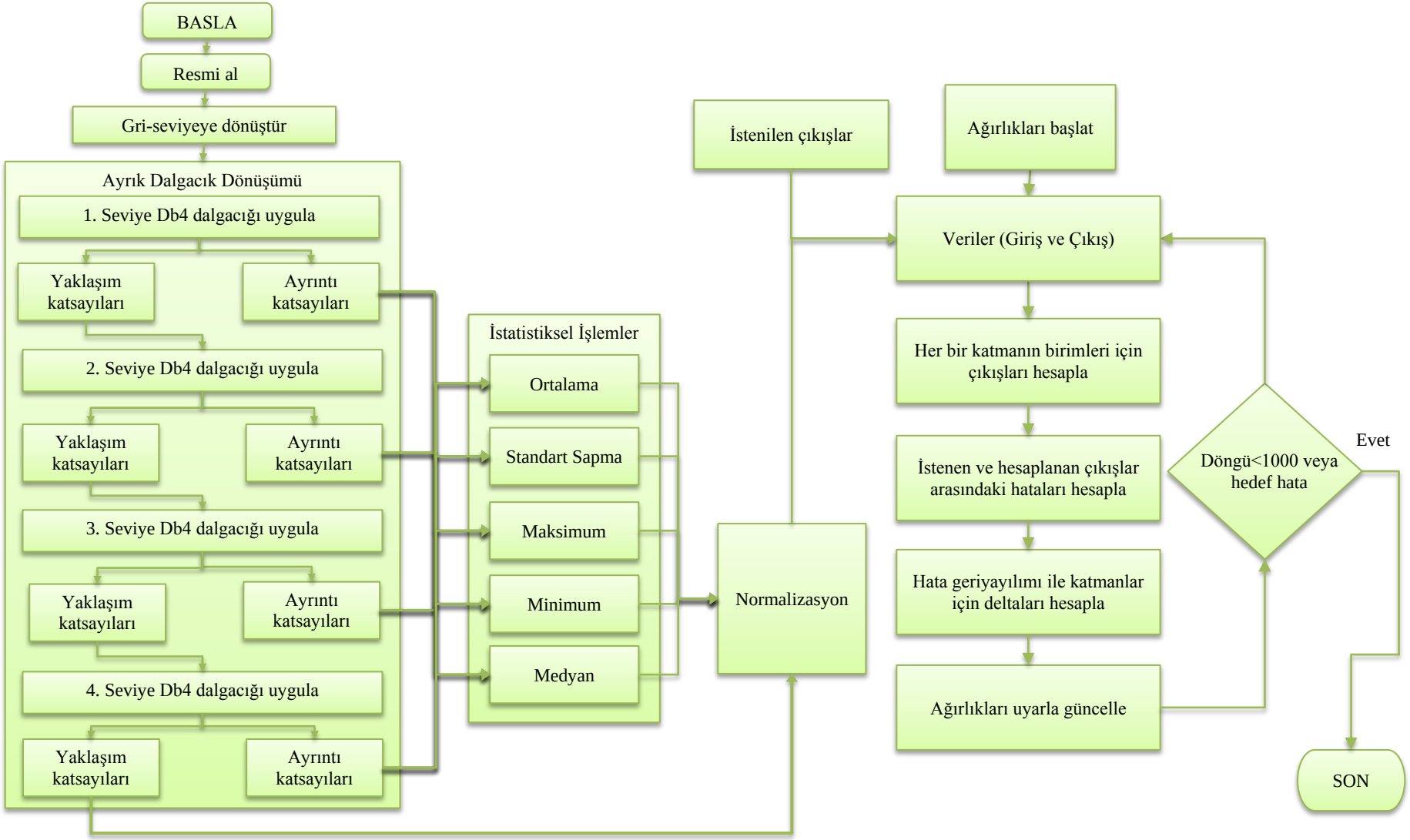
Şekil 3.17. Örüntü tanıma işlem basamakları

Görüntü İşleme: Ön işlem aşamasıdır. Görüntünün filtre edildiği, çeşitli dönüşüm ve gösterim teknikleri ile işlendiği, bileşenlerine ayrıldığı veya modellendiği kısımdır.

Özellik Çıkarma: İşaret ve görüntünün veri boyutunun indirgendiği ve tanımlayıcı anahtar özelliklerinin tespit edildiği ve aynı zamanda normalizasyona tabii tutulduğu aşamadır. Sistemin başarımında en etkili rolü oynar.

Sınıflandırma: Çıkarılan özellik kümesinin indirgendiği ve formüle edildiği tanımlayıcı karar aşamasıdır.

Bu tezde yapılan çalışma için örüntü tanıma işlem basamaklarını içeren uygulamanın detaylı algoritması Şekil 3.18'de verilmiştir.



Şekil 3.18. Geliştirilen uygulama algoritması

3.2.1 Dalgacık analizi ile kumaş bilgisinin elde edilmesi

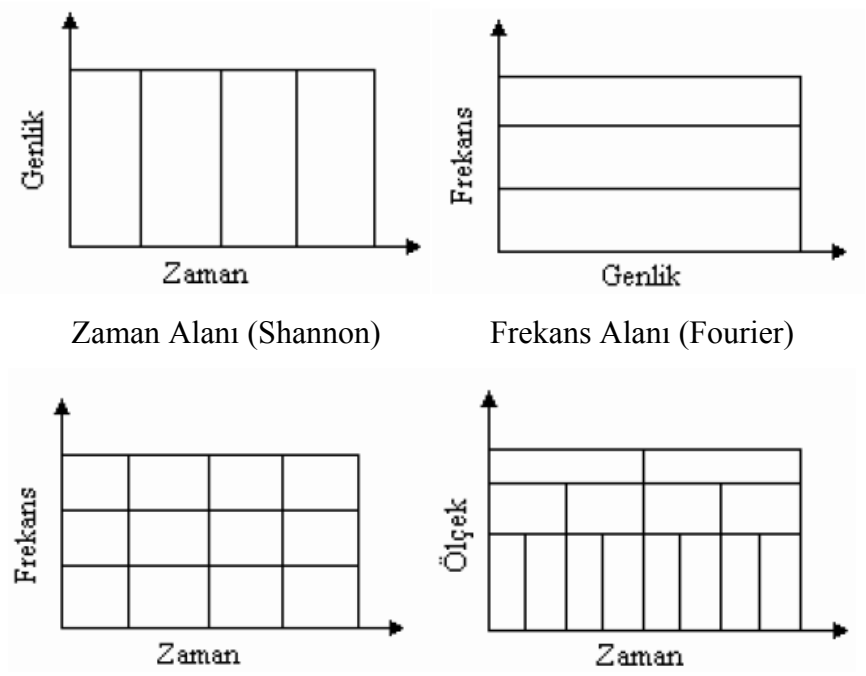
3.2.1.1 Dalgacık analizi

Sinyal analizinde kullanılan tekniklerin içerisinde en yeni ve en verimli sonuç veren dalgacık dönüşümü yaklaşımının temelleri 1909 yılında matematikçi Haar tarafından oluşturulmuştur. Paul Levy, Haar temel fonksiyonunu kullanarak parçacıkların rastlantısal hareketini incelemiştir. Paul Levy, Haar'ın temel fonksiyonlarının Fourier temel fonksiyonlarına göre daha iyi modellediğini göstermiştir (Levy ve Martin, 1979).

Dennis Gabor (1946)'ın pencereleme ile yani işareti küçük parçalara ayırarak Fourier dönüşümündeki yetersizliği gidermeye çalışmıştır. Kısa-Zaman Fourier Dönüşümü (KZFD) denilen bu yöntem işareti frekans ve zaman alanında inceler. Bu yöntem, bir olayın hangi frekansta ve ne zaman olduğu hakkında bilgi içerir. Fakat zaman alanındaki pencere boyutu değiştirilirken frekans alanındaki pencerenin sabit kalması bu yöntemin eksikliği sayılır. İşaret işleme daha esnek bir yaklaşım gerektirir (Gümüş, 2003). Durağan veya durağan olmayan bir sinyaldeki daralan ve genişleyen pencere fonksiyonlarını kullanarak, frekans bilgisi ile birlikte zaman bilgisinin de elde edilmesini sağlayan dalgacık analizinin mühendislikteki kullanımı, 1984 yılında Grossman ve J. Morlet'in çalışmalarında dalgacık dönüşümü ile sismik işaretlerinin araştırılması ile başlar. Daha sonra Stephane Mallat'ın çalışmalarında sayısal işleme için Dalgacık analize yer vermesi ile günümüze kadar gerçekleştirilen ilgili uygulamaların başını çekmiştir (Daubechies, 1990). En çok kullanılan dalgacık dönüşümü Belçikalı matematikçi Ingrid Daubechies tarafından 1988 yılında bulundu. Özellikle Daubechies, Coifman ve Wickherhouser gibi araştırmacılar dalgacık dönüşümünün gelişiminde çalışmalarıyla büyük katkı sağlayıp konuya ivme kazandırmışlardır. Aynı zamanda popüler bir sinyal işleme metodu da kazandırdılar. Sweldens, çalışmasında bir fonksiyonun dalgacık sabitlerini bu fonksiyonun örneklerinden hesaplanması üzerine iterpolasyon, quadrature formül ve filtreleme metotlarıyla uygulamalar sunmuştur (Sweldens 1994). Grap (1995), makalesinde geleneksel Fourier metotları, dalgacık teorisi ve analizlerinin gelişimini inceleyerek sinyal işleme alanı temelli çeşitli karşılaştırmalar yapmış, ayrıca henüz gelişen dalgacık analizinin uygulama alanlarından bahsetmiştir. Miner (1998), bir raporunda geleneksel Fourier metotları, dalgacık teorisi ve analizlerinin gelişimini inceleyerek sinyal işleme alanı temelli çeşitli karşılaştırmalar yapmış, ayrıca sürekli ve ayrık dalgacık analiz algoritmalarını vermiştir. Torrence ve ark. (1998), çalışmalarında yaygın olarak kullanılan

temel dalgacık fonksiyonları sınıflandırmış, özellikleri verilerek ayrık zaman dizileri için sürekli dalgacık dönüşümü yaklaşımını sunmuştur. Sonlu uzunlukta zaman serilerine pencerelenmiş Fourier ve dalgacık dönüşümü uygulayarak istatistik önem ile güvenirlik aralığı analizleri yapmışlardır.

Dalgacık paketleri analizleri, KZFD analizlerine göre çok daha uygun bir araç olarak kabul edilmektedir. KZFD analizlerinde, zaman ve frekans çözünürlüğü sağlamaktadır ancak her ikisini bir yapamamaktadır. Dalgacık paket analizinde ise zaman ve frekans çözünürlüğü istenilen seviyede elde edilmesi olası bir durum olarak göz önüne serilmektedir. Ayrıca dalgacık paket dönüşümünde, eğilim yok etme yöntemlerine ihtiyaç duyulmamaktadır (Wiklund ve ark., 1997). Teorik olarak sıfır ortalamalı ve sonlu enerjiye sahip herhangi bir fonksiyon dalgacık sayılabilir. Ancak, dalgacığı seçmek için birçok ölçüt vardır. Dalgacığın, zaman ve frekans ortamındaki sönümlenmesi önemlidir. Zaman ve frekans ortamında iyi lokalize olabilmek için dalgacık, zaman ve frekans ortamında hızlı sönümlenmelidir (Tüfekci ve Gowdy, 2000). Düşük ölçeklerde dalgacık daralır ve ani değişimler (yüksek frekanslar) yakalanır. Yüksek ölçeklerde ise dalgacık genişleyerek düşük frekanslar yakalanır. Dalgacık analizinde işaret zaman-frekans yerine zaman-ölçek ortamında incelenir ve en önemli üstünlüklerinden biri; işareti belirli bir bölgenin analizin mümkün olmasıdır (Şekil 3.19).



Zaman Alanı (Shannon) Frekans Alanı (Fourier)

KZFD (Gabor) Dalgacık Analizi

Şekil 3.19 Analiz yöntemleri arasındaki ilişki

Son yıllarda, dalgacıkları kullanarak sinyalleri analiz etmek ve uygulamalarda kullanmak oldukça yaygınlaşmıştır. Özellikle sinyal islemede daha önce kullanılan geleneksel yöntemlere göre daha avantajlı olan dalgacık analizleri, matematik, kuantum fiziği, elektrik, elektronik ve bilgisayar gibi mühendislik alanlarında yeni bir devir açma niteliğindedir. Uygulama alanları da gün geçtikçe artmaktadır (Uçan, 2003).

3.2.1.1.1 Dalgacık dönüşümünün matematiksel tabanı

Kısa zaman Fourier dönüşümünde bir pencere fonksiyonu bulunurken, dalgacık dönüşümünde bir dalgacık $\psi(x)$ fonksiyonu kullanılmaktadır. Söz konusu dalgacık fonksiyonu ölçeklendirilip ve zaman alanında kaydırılarak analiz işlemi gerçekleştirilir. Dalgacıklarda, öncelikle aşağıdaki iki şartı sağlayan bir gerçek değerli fonksiyon $\psi(x)$ olması gerekir (Percival ve Walden, 2002).

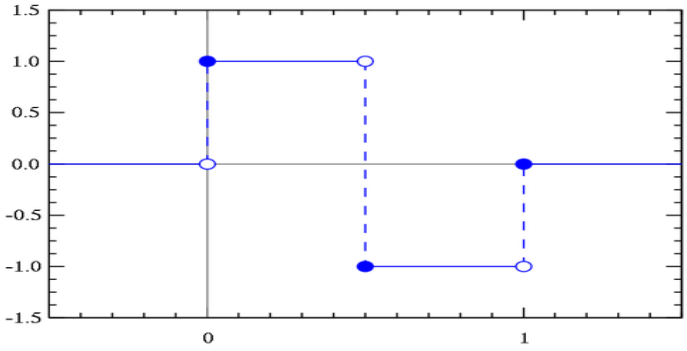
$\psi(x)$' integralinin sıfır olması $\int_{-\infty}^{\infty} \psi(x)d(x) = 0$

$\psi(x)$'nin karesinin integralinin 1 olması $\int_{-\infty}^{\infty} \psi^2(x)d(x) = 1$

Bu şartları sağlayan her $\psi(x)$ fonksiyon dalgacık olarak adlandırılır. Yukarıdaki eşitliği sağlayan en temel dalgacık fonksiyonu Haar dalgacığı olarak bilinmektedir. Matematiksel olarak şu şekilde ifade edilir;

$$\psi^{(h)}(x) \equiv \begin{cases} -1/\sqrt{2}, & -1 < x \leq 0; \\ 1/\sqrt{2}, & 0 < x \leq 1; \\ 0 & diğer\ durumda \end{cases} \tag{3.1}$$

Haar dalgacığının grafiksel gösterimi şekildeki gibidir. (Şekil 3.20)



Şekil 3.20 Haar dalgacığı

Diğer bir deyişle dalgacık dönüşümündeki temel fikir, bir f işaretini ψ_i fonksiyonuna bağlı olarak bileşenlerine ayırmaktır.

$$f = \sum_i a_i \psi_i$$

(3.2)

Buradaki ψ_i fonksiyonunun uygun dalgacık ailesinden seçilmesi çok önemlidir. İşaretle (sinyalle), ψ_i fonksiyonu birbirine uygun olmalıdır. Bir *x[n]* işaretinin ayrık dalgacık dönüşümü bu işaret üzerinden alçak geçirgenlikli *g[n]* ve yüksek geçirgenlikli *h [n]* filtresinin geçirilmesiyle hesaplanır. *g[n]* ve *h[n]* filtreleri ana dalgacık adı verilen bir fonksiyondan türetilmişlerdir (Busch, 1997).

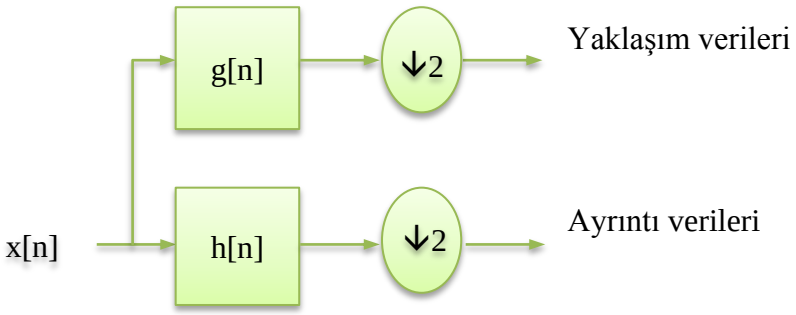
Matematiksel olarak şu şekildedir.

$$y_{alçak}[n] = \sum_{k=-\infty}^{\infty} x[k].g[2n-k]$$

$$y_{yüksek}[n] = \sum_{k=-\infty}^{\infty} x[k].h[2n-k]$$

(3.3)

Filtreleme işleminin sonucunda yüksek geçirgenlikli filtreden elde edilen verilere ayrıntı verileri, alçak geçirgenlikli filtreden elde edilen verilere ise yaklaşım verileri denmektedir.



Şekil 3.21 Filtre analizinin blok diyagramı

Dalgacık dönüşümü, sürekli ve ayrık dalgacık dönüşümü olmak üzere temel olarak 2 kısımda tanımlanır.

3.2.1.1.2 Sürekli dalgacık dönüşümü

Matematiksel olarak Fourier analiz süreci Fourier dönüşümüyle gösterilir. (Denklem 3.4)

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

(3.4)

Eşitliği *f(t)* sinyalinin bir karmaşık üstel ile çarpılıp toplanmasıyla elde edilir. Dönüşümün amacı bu Fourier katsayılarını hesaplamaktır. Böylece bir sinyal Fourier dönüşümü yardımıyla bileşenlerine ayrılır. Her bileşenin ayrı genliği ve frekansı vardır. Benzer şekilde Sürekli Dalgacık Dönüşümü(SDD), dalgacık fonksiyonunun ψ kaydırılıp bir ölçekle, çarpıldıktan sonra zaman alanı boyunca toplanmasıdır.

Bunun matematiksel ifadesi,

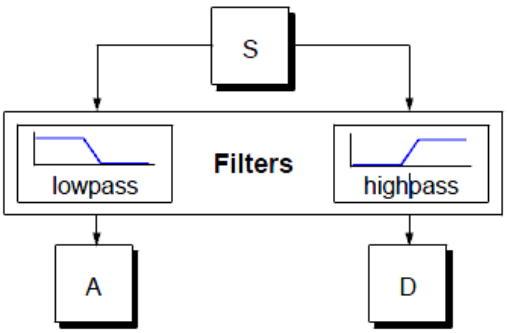
$$SDD(\tau, s) = \frac{1}{\sqrt{|s|}\int_{-\infty}^{\infty} f(t)\psi\left(\frac{t-\tau}{s}\right)dt} \tag{3.5}$$

biçimindedir. Buradaki s ölçek ve τ konumu ifade etmektedir. SDD'nin sonucunda ölçek ve konum fonksiyonu olan birçok dalgacık katsayıları C elde edilir. (yaklaşım ve ayrıntı katsayıları) Sürekli dalgacık dönüşümleri analog işlemlerde kullanılır.

3.2.1.1.3 Ayrık dalgacık dönüşümü

Ayrık dalgacık dönüşümü, mümkün olan her ölçek için dalgacık katsayılarının hesaplanması sırasında, gereksiz olan bilgilerin sinyal içerisinden alınmasından dolayı çok büyük veri yığınları oluşturmaktadır. Sadece hesap sınırı dizinin zaman alanına, ölçek ve konum değerleri de araştırmacının tercihine bağlıdır. Bu nedenle, belirli ölçekler tespit edilir ve bu aralıkta analizler yapılır. Çoğunlukla en pratik ve kullanışlı yol, ölçek ve konum değerleri 2'nin kuvveti olacak şekilde seçilmesidir. Matematiksel kuram olarak SDD'den hiç bir fark yoktur. Eğer seçilecek olan ölçek ve pozisyonları, 2'nin üstleri biçiminde seçilirse, yapılan analiz daha etkili ve daha hızlı olur. İstenmeyen verilerin analiz işleminden ayıklanmasını sağlar ve Ayrık Dalgacık Dönüşümü (ADD) olarak adlandırılır. ADD, filtreleri kullanarak gerçekleştirilmektedir. 1988 yılında Mallat tarafından gerçekleştirilen bir algoritmadır. Bu çok pratik filtreleme algoritması, dalgacık dönüşüm işleminin SDD'ye nazaran daha hızlı yapılmasını sağlar (Graps, 1995).

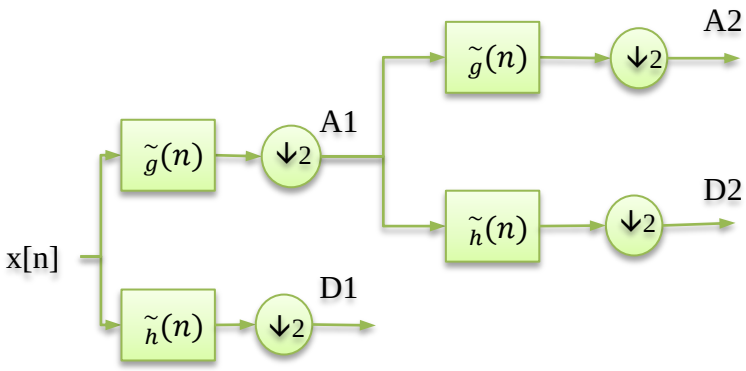
Şekil 3.22'de bir sinyale filtre uygulanışı ve elde edilen değerler verilmiştir. En temel filtre devresinde S sinyaline uygulanan bir filtre ile sinyal içerisindeki alçak ve yüksek frekanslar ayrılmıştır. Düşük frekansları geçiren filtre Alçak Geçirgen Filtre (AGF) ve yüksek frekansları geçiren filtre de Yüksek Geçirgen Filtre (YGF) olarak adlandırılır.



Şekil 3.22. Temel filtreleme uygulanması

Filtre sonucunda oluşan iki ayrı sinyalden A ile gösterilene yaklaşım ve D ile gösterilene de ayrıntı katsayıları denilir. Yaklaşım katsayıları, orijinal sinyali temsil eder ve sinyalin tanımını verir. Ayrıntı katsayıları ise sinyalin karakteristiğini yâda ayrıntısını içerir. Örnek olarak insan sesini ele alırsak; insan sesinden yüksek frekanslar kaldırılırsa konuşmanın içeriği anlaşılabilir. Ancak düşük frekanslar kaldırılırsa içeriği anlaşılmayan, anlam verilemeyen sesler duyulur. Dalgacık dönüşümünde de yaklaşımlar sinyalin orijinalini temsil eder ve yüksek ölçek ile elde edilir. Detaylar ise düşük ölçekteki bilgilerdir ve sinyalin yüksek frekanslı bileşenleridir. Sinyallere alçak geçirgen ve yüksek geçirgen filtreler uygulayarak ve ortaya çıkan verileri iki ile azaltarak bu sinyal ADD yapılabilir. Bu işlem çıkan sonuçlara da uygulanırsa çoklu çözünürlük analizi sinyale uygulanmış olur (Merry 2005).

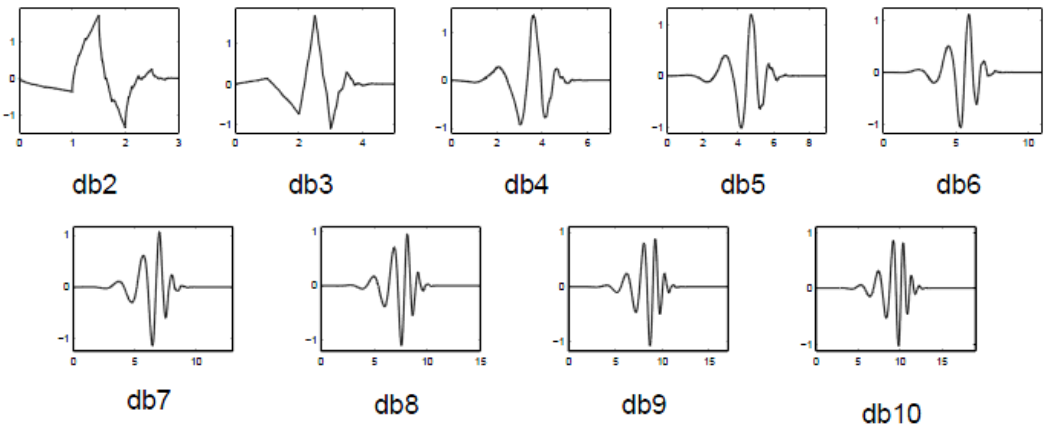
Şekil 3.23'de X(n) sinyaline uygulanan prosedür gösterilmekte ve burada YGF ve AGF sırasıyla yüksek geçirgen ve alçak geçirgen filtreleri belirtilmektedir. Aşağı doğru ok ile gösterim ise iki üssü azaltılma (Downsampling) işlemidir. Her bir ayrıştırma seviyesinde ortaya çıkan detaylar ve yaklaşımlar gösterilmektedir. X(n) sinyalinin içerisindeki bilgi A2 yaklaşımıdır. D1, D2 bu sinyalin farklı frekans değerlerindeki yüksek frekans kısımları olup, X(n) sinyali tüm detaylarla A2 yaklaşımının toplamına eşittir. Ayrıştırma işlemi sonrasında, istenen frekans aralığı seviyesindeki detay veya yaklaşım bilgisi seçilerek analiz edilebilir (Liu, 2005).



Şekil 3.23. Sinyal ayrıştırma

Şekil 3.23'te görüldüğü gibi her ayrıştırma aşamasında, orijinal sinyalden alınan örnek sayısı yarı yarıya azalmaktadır. Ayrıştırılan bu sinyallere ters ayrıştırma işlemi uygulanarak, orijinal sinyali tekrar elde etmek mümkündür. Dalgacık dönüşümlerinin bu yönü de oldukça kuvvetlidir ve görüntü islemede resim bilgisinden daha iyi bilgiler alınmasını sağladığı için mükemmel sonuçlar vermektedir (Chhokra, 2002).

Haar dalgacığı, dalgacık tipleri arasında en basit olanıdır. Haar dalgacığı sadece iki ölçekleme katsayısına sahip olup 'db1' ile gösterilir. 'db2' dalgacığı dört, 'db3' dalgacığı altı, 'dbN' dalgacığı ise, 2N adet ölçekleme katsayısına sahiptir (Daubechies,1992).Şekil 3.24'te Daubechies dalgacık grafikleri verilmiştir.



Şekil 3.24. Daubechies dalgacık ailesi

3.2.1.2 Kumaş bilgisinin elde edilmesi

Geliştirilen bu sistemde ilk olarak dijital kamera ile KİPAŞ firmasından göz ile tespit edilebilen ve hata içeren kumaş örnekleri resimlenmiştir. Alınan bu hatalı kumaş resim örnekleri Ek-1'de görülmektedir. Geliştirilen uygulama MATLAB programında bulunan görüntü işleme araç kutusu (Image Proccessing Toolbox) kullanılarak yazılım

ortamına aktarılmıştır. Bu işlem için M-dilinde aşağıdaki kod parçacığı ile gerçekleştirilmiştir. Daha sonra uygulamaya aktarılan resimler RGB modundan gri seviye resmine dönüştürülmüş ve double precision işlem hızı için double değişken tipine dönüştürülmüştür.

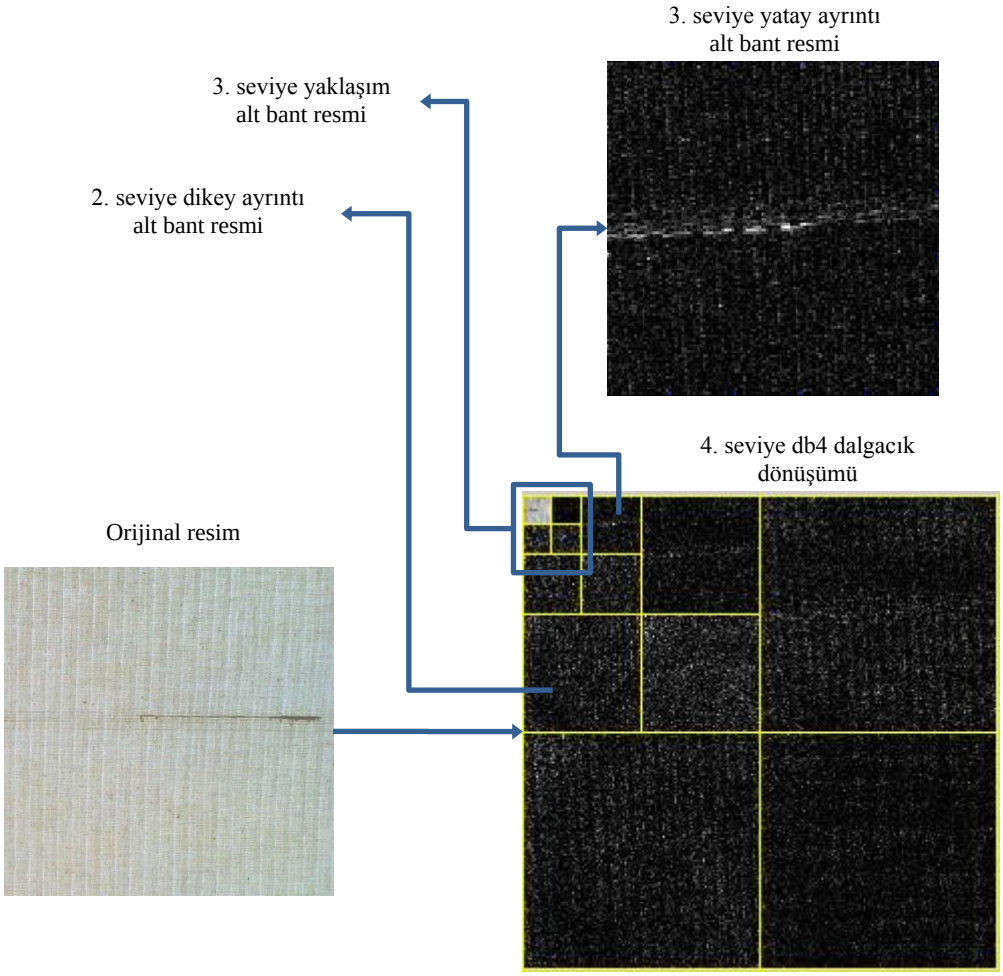
```
resim=imread('resim1.jpg']);
resim=rgb2gray(resim);
resim=gdouble(resim);
```

Jacket ara yazılımına ait gdouble değişken tanımlama fonksiyonu ile *resim* değişkeni GPU'da çalıştırılmak üzere hazırlanmıştır. Uzaysal alanda görülemeyen veya elde edilemeyen ayrıntı bilgileri için filtreleme, dönüşümler gibi yöntemlere başvurulur. Zaman-ölçek tabanında ayırt edici bilgi içeren çoklu çözünürlük analizi (dalgacık dönüşümü) kullanılarak bu resimler analiz edilmiştir.

Gri seviyeye dönüştürülmüş kumaş görüntüleri Daubechies dalgacık ailesinden db-4 dalgacığı kullanılarak 4. seviyeye kadar ayrıştırılmıştır. Bu işlem aşağıdaki kod parçası ile gerçekleştirilir.

```
[c, s]=dalgacik(resim, n, yu_g, al_g, fl)
```

Dalgacık fonksiyonu içerisinde yer alan hesaplamaların hepsi Jacket ara yazılımının desteklediği kodlar düşünülerek yazılmıştır. Dalgacık fonksiyonunda *n* (4. seviye için n=4) dalgacık dönüşümü seviyesi, *yu_g* db4 dalgacığı yüksek geçirgen filtresi, *al_g* db4 dalgacığı alçak geçirgen filtresi *fl* al_g filtresi boyutudur. *c* tüm katsayıların saklandığı değişken ve *s* ise her seviye için dönüşüm bilgilerinin saklandığı değişkendir. Resme db4 dalgacığı ile uygulanan 4. seviye dönüşümden sonra alt bant resimleri (katsayıları) elde edilir. Bunlar; yaklaşım alt resmi, yatay alt resmi, dikey alt resmi ve çapraz alt resmidir (Şekil 3.25)



Şekil 3.25 4. seviye Dalgacık dönüşümü

3.2.2 Öznitelik Çıkarımı/Seçimi

Öznitelik çıkarma, örüntü tanımlama ve örüntünün önemli özniteliklerinin çıkarılıp öznitelik vektörünün elde edilmesi işlemidir. Öznitelik seçme, isteğe bağlı olarak yapılan bir işlem olup sınıflandırma işlemi açısından en belirleyici özniteliklerin seçilmesi ile öznitelik vektörünün boyutunun azaltılmasıdır.

Dalgacık dönüşümü ile elde edilen alt bant resimleri (katsayıları) üzerinden istatistiksel işlemler uygulanarak öznitelik vektörleri çıkarımı ve seçimi yapılmıştır. Uygulamada kullanılan İstatistiksel işlemler ortalama, standart sapma, maksimum, minimum ve medyandır.

Ortalama: X, N adet örnekten oluşan veri kümesi ise şu şekilde hesaplanır**:**

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \quad X = x_1, x_2, x_3 \ \ldots . x_N \tag{3.6}$$

Standart Sapma: Standart sapma, veri kümesini oluşturan örneklerin dağılımının ölçüsü olarak tanımlanır. Standart sapma aşağıdaki gibi hesaplanır:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_n - \mu_x)^2} \tag{3.7}$$

Maksimum: Maksimum değer, veri kümesini oluşturan örneklerden en büyük olanıdır. Şu şekilde ifade edilir:

$$y = mak(X) \tag{3.8}$$

Minimum: Minimum değer, veri kümesini oluşturan örneklerden en küçük olanıdır. Şu şekilde ifade edilir:

$$y = min(X) \tag{3.9}$$

Medyan: Medyan ortanca da denilir. Veri kümesinin merkezini ölçer. Şu şekilde ifade edilir:

$$Me = \uparrow x_{\frac{n+1}{2}}\ n\ tek\ ise$$
$$Me = \frac{\left(\uparrow x_{\frac{n}{2}} + \uparrow x_{\frac{n}{2}+1}\right)}{2}\ n\ çift\ ise \tag{3.10}$$

Eğer veri kümesi içeresindeki elemanların sayısı *n* tek ise artan bir şekilde sıralanan veri kümesinin orta elemanı medyan değerini verir. Eğer *n* çift ise artan bir şekilde sıralanan veri kümesi içerisindeki orta eleman ve ondan sonraki elemanın ortalaması medyandır.

Bu işlemi gerçekleştirmek için aşağıdaki kod parçası kullanılmıştır.

```
x1=mean2(inputs);
x2=std2(inputs);
x3=max(max(inputs));
x4=min(min(inputs));
x5=median(median(inputs));
```

Yukarıda kullanılan fonksiyonlar Matlab programı içeresinde var olan hazır fonksiyonlardır. Bu kodlar aynı zamanda Jacket ara yazılımı tarafından desteklenmektedir. Örneğin, std2 iki boyutlu bir matrisin standart sapmasını bulmak için kullanılır. Bu hazır fonksiyonlar Jacket tarafından desteklediği için ayrıca bu fonksiyonlar temel kodlar

kullanılarak tekrar yazılmamıştır. *inputs* verisi dalgacık dönüşümünden elde edilen ayrıntı katsayılardır.

İstatistiksel işlemlerden elde edilen öznitelik vektörleri normalizasyon işlemine tabi tutulmuştur. Normalizasyon işlemi için oluşturulan fonksiyonu aşağıdadır.

```
dalgacık_ozNitelik=normalizasyon(inputs)
```

Fonksiyona parametre olarak gönderilen *inputs* değişkeni istatistiksel işlemlerden elde edilen öznitelik vektörleridir. *dalgacık_ozNitelik* değişkeni ise normalizasyon işleminden sonra elde edilen normalleştirilmiş veridir.

YSA eğitiminden önce, giriş verilerinin normalize edilmesi yani ölçeklenmesi yararlıdır. Bu nedenle normalizasyon işlemi ile YSA giriş verisi belirlenen bir aralığına indirgenir. Normalizasyon algoritması

$$y = (\mathrm{y}_{maks} - \mathrm{y}_{min}) \frac{(x - x_{min})}{(x_{maks} - x_{min})} + \mathrm{y}_{min} \tag{3.11}$$

şeklindedir. $\boldsymbol{y_{maks}}$ ve $\boldsymbol{y_{min}}$ istenilen aralık değerlerini belirleyen değişkenlerdir. $\boldsymbol{x_{maks}}$ ve $\boldsymbol{x_{min}}$ ise giriş verisi $\boldsymbol{x}$'in en büyük ve en küçük değerleridir.

İstatistiksel işlemlerden elde edilen normalleştirilmiş veri yani öznitelik vektörleri ileri beslemeli yapay sinir ağına giriş verisi olarak kullanılacaktır.

3.2.3 Yapay sinir ağları ile kumaş hatalarının sınıflandırılması

Yukarıda bahsi edilen görüntü işleme ve örüntü tanıma teknikleri için sınıflandırıcılara ihtiyaç vardır. Sınıflandırıcılar En yakın komşular, yapay sinir ağları ve destek vektör makinaları olmak üzere üç ana grupta toplanabilir. Yapay sinir ağları parametrik olmayan doğası karmaşık karar bölgelerini tanımlama gibi yeteneklerinden ötürü hata tespiti için en hızlı ve en esnek sınıflandırıcılardan birisi olarak düşünülebilir (Kumar, 2003)

Yapay Sinir Ağları (YSA) kavramı, insan beyninin bir temsili gibidir ve biyolojik sinir sistemi ile ilgili temellere dayanır. Yapay sinir ağları, insan beyninin çalışma işlemini ve çalışmasını taklit eder, örneklerle ve eğitim ile öğrenir. Bir bilgisayarın ya da bilgisayar denetimli bir makinenin genellikle insana özgü olduğu varsayılan akıl yürütme, anlam çıkartma, genelleme ve geçmiş deneyimlerden öğrenme gibi yüksek zihinsel süreçlere ilişkin görevleri yerine getirme yeteneği olarak tanımlanır (Nabiyev, 2010).

Bilgisayarlar çok karmaşık sayısal işlemleri anında çözümleyebilmelerine karşın, idrak etme ve deneyimlerle kazanılmış bilgileri kullanabilme noktasında yetersizdirler. Bu olayda insanı yâda insan beynini üstün kılan temel özellik, sinirsel algılayıcılar vasıtası ile kazanılmış ve görevli olarak sınıflandırılmış bilgileri kullanabilmesidir. Uzman sistemler, Bulanık Mantık, Genetik Algoritma (GA) ve Yapay Sinir Ağları (YSA) gibi yapay zekâ alt dalları özellikle son yıllarda, geniş bir araştırma ve uygulama alanı bulmaktadır (Elmas, 2007).

İlk olarak McCulloch ve Pitts (1943), tarafından basit nöronların tanıtılmasından sonra, bağlantılı model ya da paralel dağıtılmış işleme olarak da bilinen yapay sinir ağları ilgi odağı haline gelmiştir. Bu nöronlar, hesaplama görevlerini yerine getiren, biyolojik nöron modelleri ve elektronik devreler için kavramsal bileşenler olarak sunulmuştur. Bu nedenle McCulloch ve Pitts ilk yapay sinir ağlarının tasarımcıları olarak bilinirler. Hesaplama gücünün tamamında artışa neden olan çoğu basit işleme birimlerini birleştirmişlerdir. Yapay sinir ağları ile ilgili birçok önerilerde bulunmuşlardır. Bunlar, bir nöronun bir eşik seviyesine sahip olması ve bu seviye ulaşıldığı zaman nöronun ateşlenmesidir. Yapay sinir ağlarının çalışmasında halen önemli bir yöntemdir. McCulloch ve Pitts yapay sinir ağı sabit bir ağırlık kümesine sahiptir.

Hebb (1949) yapay sinir ağlarında ilk öğrenme kuralını geliştirdi. Yani; eğer iki nöron aynı anda etkin ise, bu iki nöron arasındaki güç arttırılmalıydı. 1950 ve 1960'larda çoğu araştırmacı, Block, Minsky, Papert ve Rosenblatt perceptron kavramı üzerinde çalıştılar. Bu yapay sinir ağı modeli problem çözümünde doğru ağırlıklara yakınsanmayı kanıtlanabilir hale getirdi. Perceptron modelinde kullanılan ağırlık ayarlama, Hebb tarafından kullanılan öğrenme kurallarından daha güçlü olduğu bulunmuştur. Böylece, Perceptron büyük bir heyecana neden olmuştur.

Minsky ve Papert (1969), doğrusal olarak ayrılamayan fonksiyonların perceptron tarafından öğrenilemeyeceği gibi perceptron modellerinin eksikliklerinin gösterildiği Perceptrons adlı kitaplarını yayınladıkları zaman, araştırmacılar yapay sinir ağı alanını terk etmişlerdir. Böylece, yapay sinir ağları araştırmalarına olan ilgi 1970'lerde ve 1980'lerin ortalarına kadar kaybolmuştu. Yalnızca birkaç araştırmacı çalışmalarına devam etmişlerdir. Bunlar Teuvo Kohonen, Stephen Grossberg, James Anderson and Kunihiko Fukushima'dır (Fausett, 1994).

İşlem kapasitesi artan yeni donanımlar ve geri yayılma algoritmasının keşfi gibi bazı önemli unsurlarla, 1980'lerin başlarında yapay sinir ağlarına ilgi tekrar ortaya çıkmıştır. J. Hopfield, gelişmeyle birlikte rastgele bulunan etkenlerin probleme karıştığını ve her çözüm yolunda problemin başlangıç halinde bulunduğuna işaret etmiş, bu nedenle her hesaplamanın doğru çözüm yolu için sürekli yenilenen bir mekanizmaya ihtiyaç olduğunu belirterek bu iş için geliştirdiği stratejiyi yapay sinir ağlarında kullanmıştır. Hopfield tarafından yapılan çalışmalardan sonra yapay sinir ağı önemi arttırmış ve modern YSA devri başlamıştır 1984 yılında Kohonen nöronların düzenli sıralanması eşleşme özelliği için danışmansız öğrenme ağlarını geliştirmiştir (Elmas, 2007).

Parker (1985), doğrusal olarak ayrılmayan problemleri çözebilen geri yayılmalı ağlar denilen çok katmanlı yapay sinir ağları için yeni bir öğrenme algoritması keşfetmiştir.1986'da Rumelhart, ileri beslemeli YSA modellerinde yeni öğrenme modeli olan hatanın geri yayılması algoritmasını geliştirmiştir. Bu algoritma ile daha önce yapılmış olan çalışmalarda ortaya çıkan sorunların çözülebileceğini göstermiştir (Rumelhart ve McClelland, 1986).

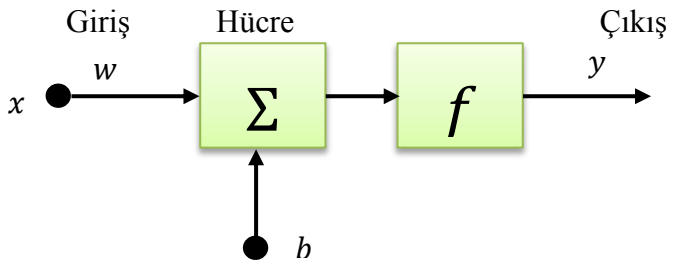
Günümüze kadar yapay sinir ağları endüstriyel, ticari ve bilimsel uygulamalarda başarı ile kullanılmıştır. Diğer taraftan da yapay sinir ağlarının kullanımı, bilim adamı sayısı, finansman miktarı, konferansların sayısı ve yapay sinir ağları ile ilgili dergilerin sayısını da etkilenmiştir. Çoğu üniversite de yapay sinir ağları ile ilgili çalışmalar sosyal bilimler, sağlık bilimleri, fen bilimleri alanlarında yapılmaktadır.

3.2.3.1 Yapay sinir ağları temel kavramları

Yapay bir sinir ağı deneysel bilgiyi depolamak ve kullanımını mevcut duruma getirmek için doğal bir eğilime sahip kitlesel paralel dağıtık bir işlemcidir. Yapay sinir ağı çok sayıda ağırlıklandırılmış bağlantılar üzerinden birbirlerine sinyal göndererek iletişim sağlayan basit işleme birimlerinden oluşur Yapay sinir ağları şu şekilde ele alınabilir; işleme birimleri (nöronlar), birimler arası iletişim ve aktivasyon (Haykin 1999).

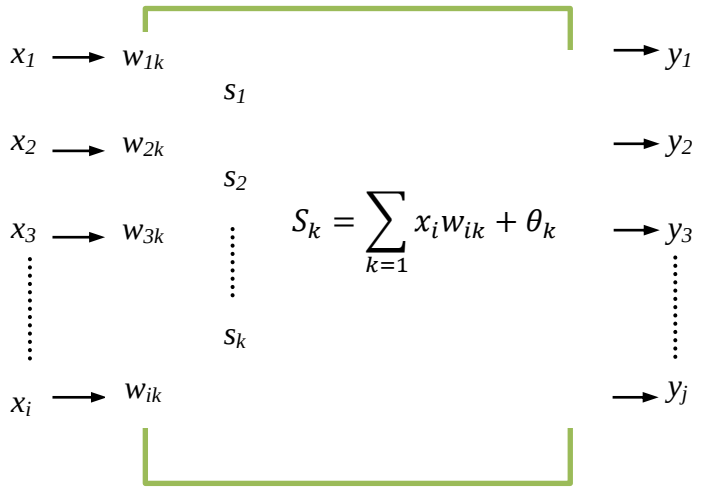
İşleme Birimleri(Nöronlar)

Yapay bir sinir ağı hücresi biyolojik sinir hücresine göre çok daha basit bir yapıya sahiptir. En temel nöron modeli Şekil 3.26'da görülmektedir.



Şekil 3.26. Temel yapay sinir ağı hücresi

Yapay sinir ağı hücresinde temel olarak dış ortamdan ya da diğer nöronlardan alınan veriler; girişler (x_i), ağırlıklar (w_i) , toplama fonksiyonu (v_i), aktivasyon (etkinlik) fonksiyonu ve çıkışlar (y_i) bulunmaktadır. Her bir nöron aslında dışarıdan ya da diğer nörondan aldığı verileri benzer şekilde işleyerek diğer nöronlara aktaran bir çıkış sinyali hesaplar. Bu işlem sürecinde diğer görev ise ağırlıkların uyarlanmasıdır. Sistem paralel çalışmaktadır ve aynı katmandaki nöronlar eş zamanlı hesaplamaları gerçekleştirir. Yapay sinir ağlarındaki birimler üç kısımda incelenebilir. Bunlar, ***i*** ile indekslenmiş ve verileri yapay sinir ağının dışındaki bir kaynaktan alan giriş birimleri, ***j*** ile indekslenmiş ve yapay sinir ağındaki çıkışı veren çıkış birimi, ***k*** ile indekslenmiş ve işlenmiş giriş ve çıkış verilerinin tutulduğu gizli birimleridir.



Şekil 3.27 Temel yapay sinir ağı hücresi

Şekil 3.27'de x_1, x_2, ... x_i giriş birimlerini, s_1, s_2, ……. s_k gizli birimlerini y_1 ,y_2 .... ,y_j çıkış birimlerini w, θ_k her bir bağlantı için ağırlıkları ve eşik değerini a_j aktivasyon fonksiyonunu göstermektedir.

Nöronlar Arası İletişim

Her bir nöron bağlantılı olduğu diğer nöronun giriş verisine toplama işlemi ile katkı sağlar. Bir nöron için giriş verisi bağlantılı bulunduğu diğer nöronlardan gelen ağırlıklandırılmış çıkış verileri ile eklenen eşik değeri toplamıdır:

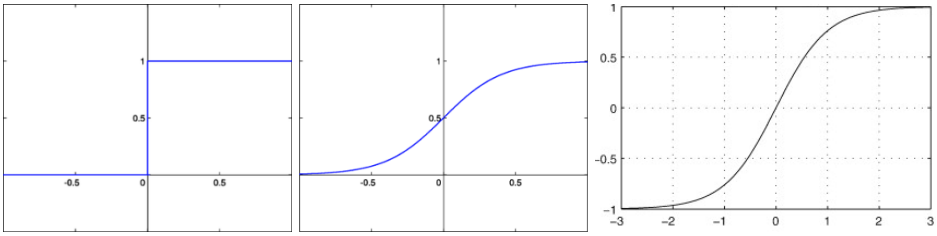
$$S_k = \sum_{k=1} x_i w_{ik} + \theta_k \tag{3.12}$$

Aktivasyon

Her bir nöron veya birimin aktivasyonu üzerinde giriş verisinin etkisini veren bir kural vardır.

$$y_j\ (t+1) = a_k, (\ y_j(t), s_k(t)) \tag{3.13}$$

Burada a_k aktivasyon fonksiyonu ile en son giriş değeri s_k ve en son aktivasyon y_j, k'nıncı nöronun yeni bir aktivasyon değerini üretir. Nörona gelen verinin aktivasyon fonksiyonu ile işlenip gelen veriye karşılık üretilen çıktı değeri aktivasyon fonksiyonlarına göre farklılık gösterir. Yani aktivasyon fonksiyonu olarak farklı tipteki aktivasyon fonksiyonlarından birisi kullanılabilir. Şekil 3.28'de bazı kullanılan aktivasyon fonksiyonları örnekleri gösterilmiştir.



Şekil 3.28. Sırasıyla Eşik, Sigmoid ve Hiperbolik Tanjant aktivasyon fonksiyonu

Eşik aktivasyon fonksiyonu ile nörondan elde edilecek çıkış değeri aralığı [-1,0] dır. Şu şekilde formülize edilir.

$$f(x) = \begin{cases} 1, x \geq 0 \\ 0, x \leq 0 \end{cases} \qquad f(x) \in \{0,1\} \tag{3.14}$$

Sigmoid fonksiyonu $(-\infty, +\infty)$ aralığında yer alan herhangi bir değeri (0,1) aralığındaki değere dönüştürür. Sigmoid aktivasyon fonksiyonu:

$$f(x) = \frac{1}{1+e^{-x}} ve\ f(x) \in (0,1) \tag{3.15}$$

Hiperbolik tanjant fonksiyonu, aynı aralıktaki değeri (-1,1) aralığındaki değere dönüştürür. Hiperbolik tanjant aktivasyon fonksiyonu:

$$f(x) = \frac{e^{2x} + 1}{e^{2x} - 1} ve\ f(x) \in (-1,1) \tag{3.16}$$

Tüm çıkış birimlerinin değerinin (0,1) aralığında ve toplamlarının da 1 olmasını sağlar.

$$y_i = \frac{e^{x_i}}{\sum_{j=1}^{c} e^{x_j}} \tag{3.17}$$

$\boldsymbol{y_i}$ çıkış biriminin değeri, $\boldsymbol{x}$ çıkış birimine gelen ağ girişleri, $\boldsymbol{c}$ çıkış birimleri sayısıdır. Kategorik hedef değişkenleri için yapay sinir ağının çıkış katmanına softmax aktivasyon fonksiyonu atanarak çıkışlar sonraki olasılık için yorumlanabilir. Bu özellik sınıflandırmada kesin ölçümler verebildiği için sınıflandırma işleminde tercih edilmektedir. Aktivasyon fonksiyonlarının seçimi yapay sinir ağlarının uygulama alanlarına göre belirlenebilir.

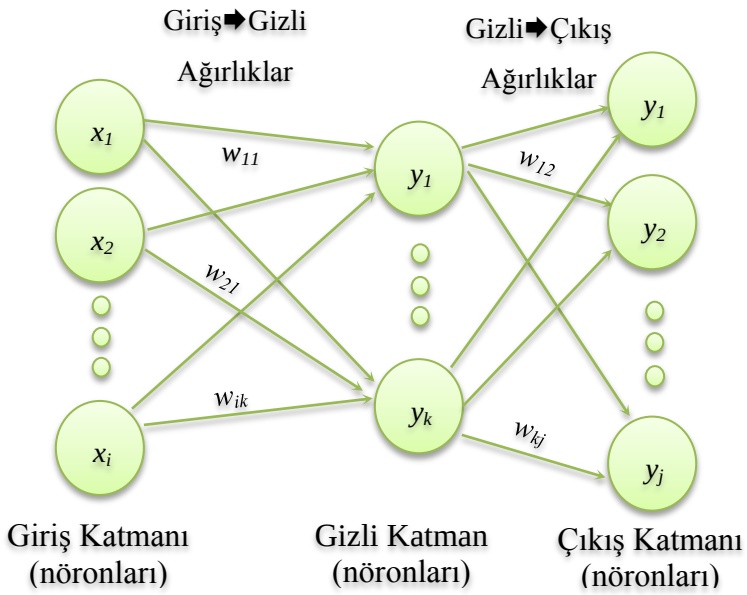
3.2.3.2 Ağ topolojileri

Nöronlar bir araya gelerek yapay sinir ağını oluştururlar. Yapay sinir ağı yapısını nöronların sayısı, nöronların örüntüsü ve bağlantı tipleri belirler. Ama genel olarak yapay sinir ağları bağlantı örüntülerine göre iki ana grupta toplanabilir. Bunlar; ileri beslemeli ağlar ve geri beslemeli (yinelenen) ağlardır.

İleri beslemeli ağlar

İleri beslemeli ağlarda, veri aktarımı giriş birimlerinden (nöronlardan) çıkış birimlerine olacak şekilde ileri doğrudur. Şekil 3.29'da görüldüğü gibi nöronların çıkışlarından aynı katmanda veya önceki katmanlarda yer alan nöronların girişlerine bağlantı yapılmasına izin verilmez. Bu topoloji, yapay sinir ağına problemle ilgili tüm bilgilerin bir kerede giriş olarak verilebileceği, çıktının tamamen o andaki girdi değerlerine bağlı olduğu durumlarda kullanılır. İleri beslemeli ağlara örnek olarak perceptron ve adaline verilebilir. Örüntü tanıma, veri sınıflandırma gibi uygulamalarda çok katmanlı ileri beslemeli yapay sinir ağları kullanılabilir. Çok katmanlı ileri beslemeli ağlar giriş katmanı, çıkış katmanı, bir veya birden fazla ara katmanları içeren bir yapıya sahiptir. Ara katmanlara gizli katmanlar denilir. İleri beslemeli ağlara örnek olarak çok katmanlı

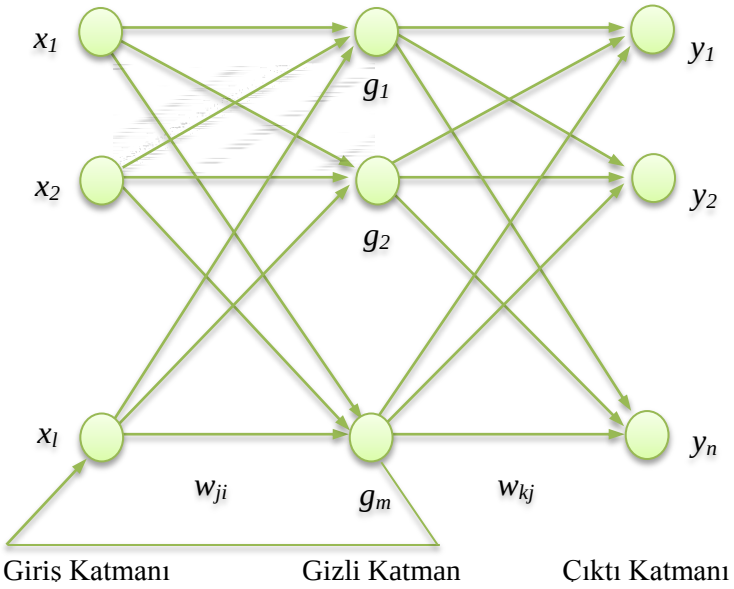
algılayıcı (Multi Layer Perceptron-MLP) ve LVQ (Learning Vector Quantization) ağları verilebilir.



Şekil 3.29. Çok katmanlı ileri beslemeli ağ modeli

Geri beslemeli (yinelenen) ağlar

Geri beslemeli ağlar, ileri beslemeli ağların tersine dinamik bir yapıya sahiptir. Bunun anlamı; geri besleme bağlantısı ile bir nörondan elde edilen çıktı tekrar başka bir nörona giriş olarak verilebilmesidir. Bu tip ağlarda giriş değerleri hem ileri hem de geri yönde aktarılabilir. Bu ağlarda, nöronların aktivasyon değerlerinin değişmeyeceği kararlı bir durum olması için aktivasyon değerleri bir rahatlatma işlemine (dinlenme süreci) tabi tutulur. Dinamik davranışların ağın çıktısını üretebilmesi açısından, uygulamalarda çıkış nöronlarının aktivasyon değerlerindeki değişim önem arz etmektedir. Geri beslemeli ağlara örnek olarak Hopfield, SOM (Self Organizing Map), Elman ve Jordan ağları verilebilir (Pearlmutter, 1990).



Şekil 3.30. Geri beslemeli yapay sinir ağı topolojisi

Geri beslemeli ağlar (Şekil 3.30), kısmi geri beslemeli ve tam geri beslemeli ağlar olarak 2 grupta sınıflandırılabilir. Kısmi geri beslemeli ağlarda, ara katman birimlerinin geçmiş durumlarını hatırlamak için kullanılan içerik birimleri vardır. Geri besleme sadece içerik birimleri üzerinden yapılır ve bu bağlantılar eğitilmezler. Fakat ağın çıktı sonuçları hem önceki durumlara hem de o anki durumlara bağlı olarak oluşmaktadır. Bu tip ağlar önceki durumları hatırlayabildikleri için dinamik bir belleğe sahiptir. Geri beslemeli ağlar için, Elman, ara katmandaki nöronlardan, ek giriş verileri olarak kullanılacak içerik elemanlarına geri besleme yapılmasını önermiştir (Elman, 1990). Jordan ise, çıkış elemanlarından içerik elemanlarına geri besleme yapan bir ağ tanımlamıştır (Jordan 1986).

Kısmi geri beslemeli ağdan farklı olarak tam geri beslemeli ağlarda, ileri ve geri besleme yapan eğitilebilir rasgele bağlantılar vardır. Bazı durumlarda, birimlerin aktivasyon değerleri bir dinlenme sürecine girer ve ağ aktivasyonların değişmediği kararlı bir duruma geçer. Ancak, ağın kararlı duruma geçmediği durumlar için bağlantıların ağırlıklarına bazı kısıtlamalar getirilerek ağın kararlı duruma geçmesi sağlanabilir. Bu tip ağlar ise genelde optimizasyon problemlerinde kullanılır.

3.2.3.3 Yapay sinir ağlarında öğrenme ve eğitim

Yapay sinir ağında öğrenme, istenilen işlevin gerçekleştirilebilmesi için ağırlıkların ayarlanma sürecidir. Ağırlıkların ayarlanmasında çeşitli yöntemler vardır. Bunlardan birisi önceki bilgileri kullanarak ağırlıkların açıkça verilmesidir. Diğeri ise nöronlar arasındaki

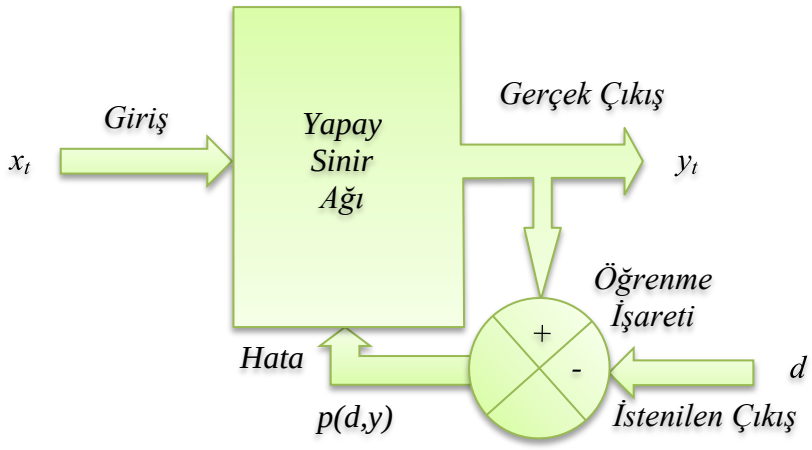
bağlantıların ağırlıklarının öğrenme kuralları dâhilinde değiştirilmesiyle ağların eğitilmesi, istenilen çıkış değerlerinin elde edilmesiyle öğrenmenin gerçekleşmesidir.

Öğrenme türleri

Temel olarak yapay sinir ağlarında öğrenme bir süreçtir. Sinaptik ağırlıklar ve eşik değerleri gibi parametrelerle yapay sinir ağı öğrenme işlemi için uyarlanır. Parametrelerdeki değişimler öğrenme tipini belirler. Bu bağlamda öğrenme süreci üçe ayrılır; danışmanlı (supervised) öğrenme ve danışmansız (unsupervised) öğrenme, donatılı (reinforced) öğrenme. Bu üç tip öğrenme temel olarak bir öğreticinin (danışmanın) var olması ya da olmamasına ve ağ öğrenmesi için sağlanan bilgilere bağlıdır.

Danışmanlı öğrenme

Öğrenme işlemi sürecinde bir danışman vardır ve beklenen çıktılar danışman tarafından sağlanır. Her bir giriş örüntüsü ağın eğitilmesinde kullanılır. Öğrenme işlemi ağın hesaplanan çıktısı ve beklenen çıktı arasındaki karşılaştırma yani karşılaştırma sonucunda elde edilen hata değerine bağlıdır. Elde edilen hataya bağlı olarak ağın performansını arttırmak için ağ parametreleri değiştirilir. Widrow-Hoff tarafından geliştirilen delta kuralı, Rumelhart ve McClelland tarafından geliştirilen genelleştirilmiş delta kuralı ve geri besleme algoritması danışmanlı öğrenme algoritmalarına örnek olarak verilebilir (Şekil 3.31) (Sağıroğlu, 2003).

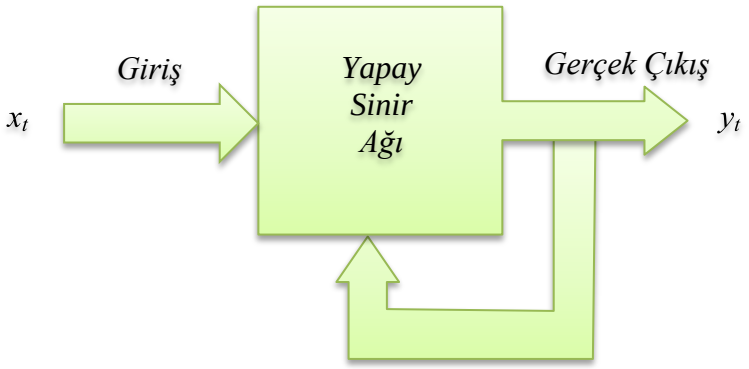


Şekil 3.31. Danışmanlı öğrenme yapısı

Danışmansız öğrenme

Kendi kendine öğrenilebilen ağlardır. Öğrenme işlemi sürecinde herhangi bir danışman yoktur ve istenilen veya beklenen çıktılar ağa sağlanmaz. Ağ, giriş örüntüsündeki yapısal özellikleri keşfederek ve kendisini uyarlayarak öğrenmeyi

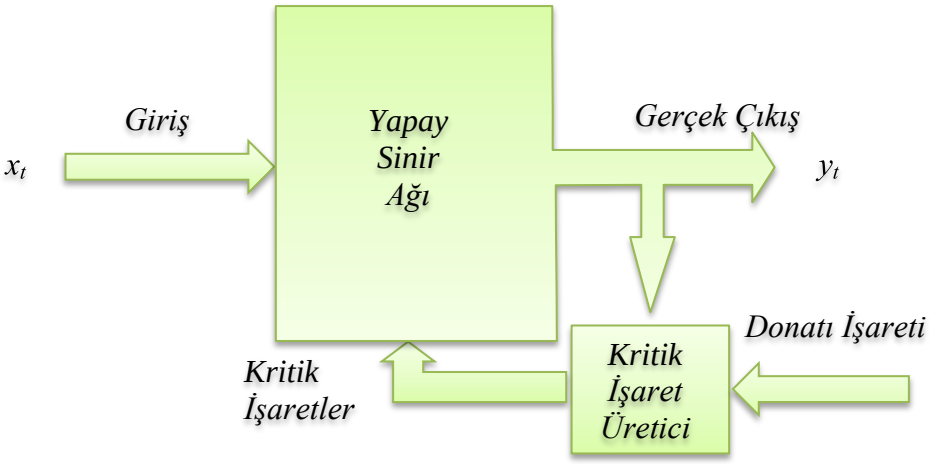
gerçekleştirir. Grossberg tarafından geliştirilen ART (Adaptive Rezonance Theory) veya Kohonen tarafından geliştirilen SOM öğrenme kuralı danışmansız öğrenmeye örnek olarak verilebilir. Şekil 3.32'de danışmansız öğrenme yapısı gösterilmiştir (Sağıroğlu, 2003).



Şekil 3.32. Danışmansız öğrenme yapısı

Donatılı (reinforced) öğrenme

Öğrenme işlemi sürecinde bir danışman vardır ve istenilen veya beklenen çıktılar danışman tarafından sağlanmaz ancak sadece hesaplanan çıktının doğru olup olmadığı bilgisi elde edilir. Öğrenme sürecinde ağa sağlanan bilgiler yardımcı olur. Hesaplanan doğru yanıt için bir ödül verilir ve verilen yanlış cevap için bir ceza verilir. Genetik algoritma, donatılı öğrenmeye örnek verilebilir. Şekil 3.33'te donatılı öğrenme yapısı verilmiştir (Sağıroğlu, 2003).



Şekil 3.33. Donatılı Öğrenme Yapısı

3.2.3.4 Öğrenme algoritmaları

Yapay sinir ağının mimarisine, karşılaşılan sorunun niteliğine göre farklılık gösteren birçok öğrenme algoritması bulunmaktadır. Bunların büyük çoğunluğu matematik

tabanlıdır ve ağırlıkların güncelleştirilmesi için kullanılırlar. Mevcut öğrenme algoritmaları Hebb, Delta, Kohonen ve Hopfield olmak üzere 4 öğrenme algoritması temel alınarak geliştirilmiştir. (Sağıroğlu ve ark. 2003, Haykin 1999).

Hebb kuralı

1949 yılında Kanadalı psikolog Donald Hebb tarafından biyolojik temele dayalı olarak geliştirilmiş olan Hebb algoritması en eski ve en ünlü öğrenme algoritmasıdır. Bu öğrenme algoritması basit bir mantığa dayanmaktadır: Eğer nöron (A) başka bir nöron'dan (B) girdi alıyorsa ve her ikisi de aktifse, (A) ve (B) arasındaki ağırlık artar. Bu düşüncenin en çok kullanılan şekli:

$$\Delta w_{jk} = a.y_j x_k$$

$$\Delta w_{jk} = \Delta w_{jk}^{(t)} = \left| w_{jk}^{(t+1)} - w_{jk}^{(t)} \; ve \; \Delta > 0' dır. \; (t = zaman) \right. \tag{3.18}$$

Bu formülde w_{jk} nöron u_k' den nöron u_j' ya olan ağırlık, y_j, u_j nöronun çıktısı ve x_k ise u_k nöronun çıktısıdır. ***a*** öğrenme katsayısı veya öğrenme oranı olarak adlandırılır ve birçok öğrenme algoritması tarafından kullanılır. Öğrenme katsayısı "0" ile "1" arasında bir değer alır ve bu değer ağın öğrenme hızını belirler. Büyük değerler daha hızlı öğrenme, küçük değerleri için daha yavaş öğrenme gerçekleşmektedir. Ancak hızlı öğrenme ağın "genelleme" yeteneğini azaltır. Genelleme yeteneği ağın eksik ve gürültülü verilerle doğru sonuçlar üretebilmesi için oldukça önemlidir. Hebb algoritmasıyla ilgili bir diğer konuda ağın eğitimden önce ağırlıklarının 0 olması gerektiğidir. (Elmas , 2007)

Hopfield kuralı

Hebb kuralına benzer ancak, kuvvetlendirme veya zayıflatmanın miktarı da belirlenir. Bu kural, her iki hücre de aktif veya her iki hücre de pasif ise bağlantı ağırlığının öğrenme katsayısı (oranı) kadar artırılması ya da azaltılmasını ister. Öğrenme katsayısı genellikle kullanıcı tarafından belirlenen 0–1 arasında sabit bir değerdir. (Elmas , 2007)

Delta kuralı

Bu kural, Hebb kuralının biraz daha geliştirilmiş ve oldukça yaygın olarak kullanılan şeklidir. Delta kuralı, işlem biriminin çıktısı ile hedeflenen çıktı değeri arasındaki farkı (delta) azaltacak yönde, giriş bağlantılarının ağırlıklarının sürekli değiştirilmesi fikrine dayanır. Bu kural, bağlantı ağırlıklarını, ağın hata kareleri ortalama değerini düşürecek şekilde değiştirir. Aynı zamanda, Widrow-Hoff öğrenme kuralı veya en küçük ortalama kareler (LMS) öğrenme kuralı olarak da adlandırılır. Delta kuralının

çalışmasında, çıktı katmanındaki delta hatası, aktivasyon fonksiyonunun türevi ile dönüştürüldükten sonra, bir önceki katmandaki giriş bağlantı ağırlıklarının ayarlanmasında kullanılır. Diğer bir deyişle, her seferinde bir katmana olmak üzere, hata önceki katmanlara geri yayılır ve ilk katmana ulaşılana dek bu geri yayma devam eder. İleri beslemeli, geri yayılımlı yapay sinir ağları, ismini, bu şekilde hata terimi hesaplanmasından almıştır. Delta kuralı kullanılırken, ağın istenen doğruluk noktasına yakınsayabilmesi için, giriş verilerinin rasgele olmasına dikkat edilmelidir (Elmas , 2007).

Kohonen kuralı

Teuvo Kohonen (Kohonen, 1988, 1995), bu kuralın geliştirilmesinde, biyolojik sistemlerin öğrenmesinden esinlenmiştir. Bu yöntemde, işlem elemanları ağırlıklarını değiştirmek ve öğrenmek için birbirleri ile yarışırlar. En büyük çıktıyı üreten hücre, kazanan hücre olarak yakınındaki hücrelerden daha kuvvetli hale gelir ve hem kazanan hücrelerin, hem de komşusu olan diğer hücrelerin ağırlıklarını değiştirebilir. Komşuluğun sınırları eğitim süresince değişim gösterebilir, genellikle büyük bir komşuluk ile başlanarak eğitim süreci ilerledikçe komşuluk sınırı daraltılır. Bu yöntem, verilerin istatistiksel ve topolojik modellemesinde kullanılır ve kendini örgütleyen yapılar veya topolojiler olarak da adlandırılırlar (Elmas , 2007).

3.2.3.5 Performans fonksiyonun seçimi

YSA'nın öğrenme performansını ölçmede performans fonksiyonu kullanılan bir ölçüttür. İleri beslemeli ağlarda birçok performans fonksiyonu kullanılmaktadır. Hata kareleri ortalaması (Mean Square Error - MSE) bunlardan bir tanesidir. MSE fonksiyonu istenen sonuç ile hesaplanan sistem çıkışı arasındaki farkın kareleri toplamının ortalaması olup aşağıdaki formülle hesaplanır:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(d_i - y_i)^2 \tag{3.19}$$

Burada d_i, istenen çıkış değerini, y_i ise YSA tarafından hesaplanan çıkış değerini ifade eder. N ise çıkış hücre sayısını belirtmektedir. Hata kareleri ortalamasının sıfıra yaklaşması istenen değere yakın çıkış değerinin elde edilmiş olması demektir.

İleri beslemeli ağlar için performans fonksiyonu olarak kullanılan diğer bir fonksiyon hata kareleri toplamı (Sum Square Error - SSE)'dır. Aşağıdaki verilen denklem kullanılarak Hata kareleri toplamı şu şekilde hesaplanır:

$$SSE = \sum_{i=1}^{N} (d_i - y_i)^2$$

(3.20)

Bunlardan başka performans fonksiyonu olarak kullanılan bir diğer hata hesaplama yöntemi hata kareleri ortalaması karekökü (Root Mean Square - RMS) fonksiyonudur ve aşağıdaki formül ile hesaplanır:

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (d_i - y_i)^2}$$

(3.21)

3.2.3.6 Çalışmada kullanılan YSA modelinin oluşturulması

Uygulama için seçilecek YSA türünün başarısı, uygulanacak olan yaklaşımlar ve deneyimlerle yakından ilişkilidir. Uygulamanın başarısında uygun yöntemi belirlemek önem arz etmektedir. YSA'nın tasarlanması için ağın yapısı, işleyişi ile ilgili aşağıda sıralanan kararların verilmesi gerekir:

- Ağ yapı özelliklerinin belirleme (katman sayısı, nöron sayıları gibi)
- Ağın işleyişi etkileyen fonksiyonları belirleme
- Ağ parametrelerini ve öğrenme algoritmasını belirleme,
- Eğitim ve test verisini oluşturma

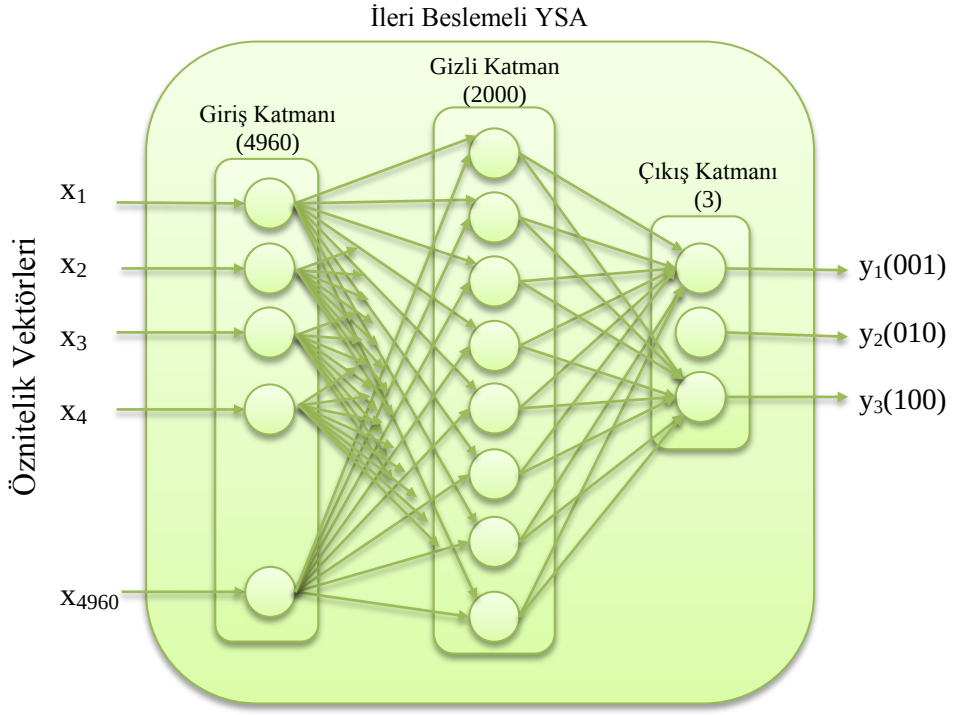
YSA'nın tasarlanması sürecinde ağ yapısı, probleme bağlı olarak seçilmelidir. Problem için hangi ağın daha uygun olduğu bilmelidir. Uygun YSA yapısının seçimi, büyük ölçüde ağda kullanılması düşünülen öğrenme algoritmasına da bağlıdır.

Bu tez çalışmasında örüntü tanıma işlemin sınıflandırma aşaması için literatürde özellikle sınıflandırma, tanıma ve genelleme yapmayı gerektiren problemlerin çözümünde kullanılan ileri beslemeli ve geri yayılmalı yapay sinir ağı kullanılmıştır. Bu YSA modelinin özellikleri Tablo 3.3'te verilmiştir.

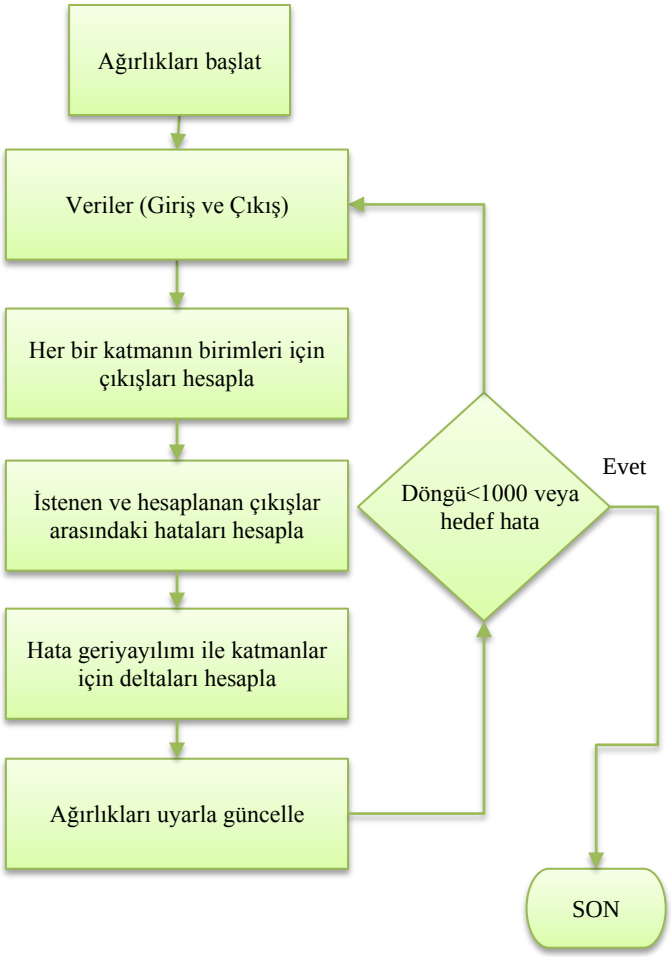
Tablo 3.3. Çalışmada kullanılan YSA modelinin özellikleri

Giriş Katman Hücre Sayısı	4960
Gizli Katman Hücre Sayısı	2000
Öğrenme Oranı	0,35
Döngü Sayısı	1000
Hata Hedefi	0,01
Performans Fonksiyonu	Hata kareleri toplamı (SSE)
Gizli Katman Aktivasyon Fonksiyonu	Sigmoid aktivasyon fonksiyonu
Çıkış Katman Aktivasyon Fonksiyonu	Softmax aktivasyon fonksiyonu

Sınıflandırmada kullanılmak üzere geliştirilen çok katmanlı ileri beslemeli YSA modeli Şekil 3.34'te verilmiştir.



Şekil 3.34. Çalışmada kullanılan çok katmanlı ileri beslemeli ağ modeli



Şekil 3.35. Çalışmada kullanılan YSA algoritması

Tasarlanan YSA algoritması Matlab'da M-dilinde yazılmıştır. YSA'da ilk olarak ağırlıkların (1 0) aralığında başlatılmalıdır. Bu işlem aşağıdaki kod parçası ile gerçekleştirilmiştir.

```
% Ağırlıkları 1-0 arasında başlatılması

% Giriş-Gizli katman arası ağırlıklar
agirlik.gir2giz =grand( kaTman.gizKatSay, gir_S+1, 'double');

% Gizli-Çıkış katman arası ağırlıklar
agirlik.giz2cik = grand( cik_S,kaTman.gizKatSay+1, 'double' );
```

grand fonksiyonu Jacket ara yazılımına ait olup Matlab'daki *rand* fonksiyonu ile aynı işlemi gerçekleştirmektedir. Yukarıdaki kod bloğunda *agirlik.gir2giz* giriş katman ve gizli katman arasındaki ağırlıkları temsil etmektedir. *kaTman.gizKatSay* değişkeni gizli katman hücre sayısını (=2000), *gir_S* (=4960) giriş katman hücre sayısını vermektedir. Böylece *grand* fonksiyonu ile 2000×4961 boyutunda rastgele (0 1) aralığında değer

kümesine sahip bir matris oluşturulur ve bu matris giriş katman gizli katman arasındaki ileri yön ve geriyayılım hesaplamalarında ağırlıklar olarak kullanılacaktır. Kod bloğundaki *gir_S+1*yazan kod için +1 ağırlıklara eklenecek eşiği (bias) ifade etmektedir. Aynı işlem sonraki kod satırında gizli katman ile çıkış katman arasındaki ağırlıkların oluşturulması için yapılmıştır. Daha sonra tüm ağ parametreleri oluşturulduktan sonra istenilen (hedef) çıktı ve örüntü ağın ileri beslemesi için ağa uygulanır. Hata hesaplamasından sonra ağ geriyayılım algoritması ile güncellenir. Bu işlemleri gerçekleştiren kod bloğu aşağıdaki gibidir.

```
while ((hkt_temp > hedHata) || (bIter < kaTman.iter))

   % İleribesleme
    [cikti, gk_Cikti] = agIleri(egit_Gir_esikli, agirlik.gir2giz,
agirlik.giz2cik, kaTman);

    % İstenilen çıktı ile ağ çıktısı arasındaki hata
    hesapHata = egit_Cik - cikti;

    % Geriyayılım ile ağırlıkların güncellenmesi
    agirlik = geriYayilim( egit_Gir_esikli, agirlik, hesapHata, gk_Cikti,
kaTman);
```

Yukarıdaki kod bloğunda *agIleri* fonkisyonu ile örüntü, ağırlıklar ve hedef çıktılar ağa uygulanarak ileri besleme işlemi gerçekleştirilir. Daha sonra gerçek çıktı ile hedef çıktı arasındaki hata hesaplaması gerçekleştirilir. Geriyayılım algoritması ile katmanlar arasındaki ağırlıklar güncellenir. Ağda hesaplanan hata (*hkt_temp*) hedeflenen hataya (*hedHata*) yaklaşınca kadar veya belirlenen ağ döngü sayısına (*kaTman.iter*) ulaşıncaya kadar eğitime devam edilir.

```
ag_cikis=agIleri(egit_Gir_esikli, agirlik.gir2giz, agirlik.giz2cik,
kaTman);
```

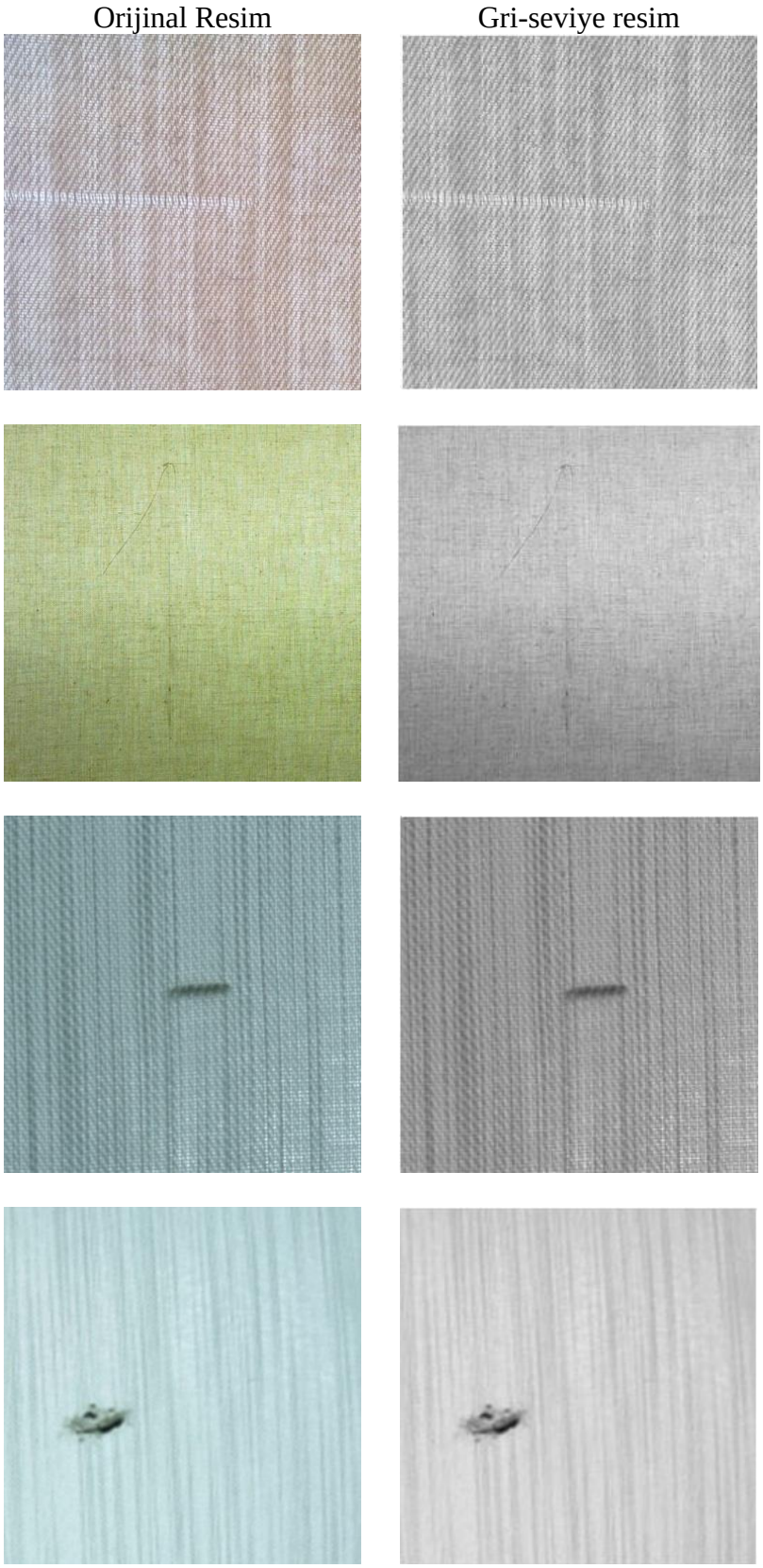
Eğitim tamamlandıktan sonra ağ *agIleri* fonksiyonu ile test edilir. Ağ testi sonuçları ile de ağ başarı düzeyi ölçülür.

4. BULGULAR ve TARTIŞMA

4.1 Öznitelik vektörleri elde etme süreci için CPU ve GPU performans karşılaştırmaları

Bu çalışmada, kumaş hata tespiti ve sınıflandırması için M- programlama dili kullanılarak, hem CPU hem de GPU üzerinde çalışan uygulama geliştirilmiştir. CPU ve GPU karşılaştırılarak aralarındaki performans değerlendirmesi yapılmıştır. Kipaş firmasından edinilen hatalı kumaş görüntüleri üzerinden örüntü tanıma yöntemiyle hata sınıflandırması yapılmıştır (Ek-1).

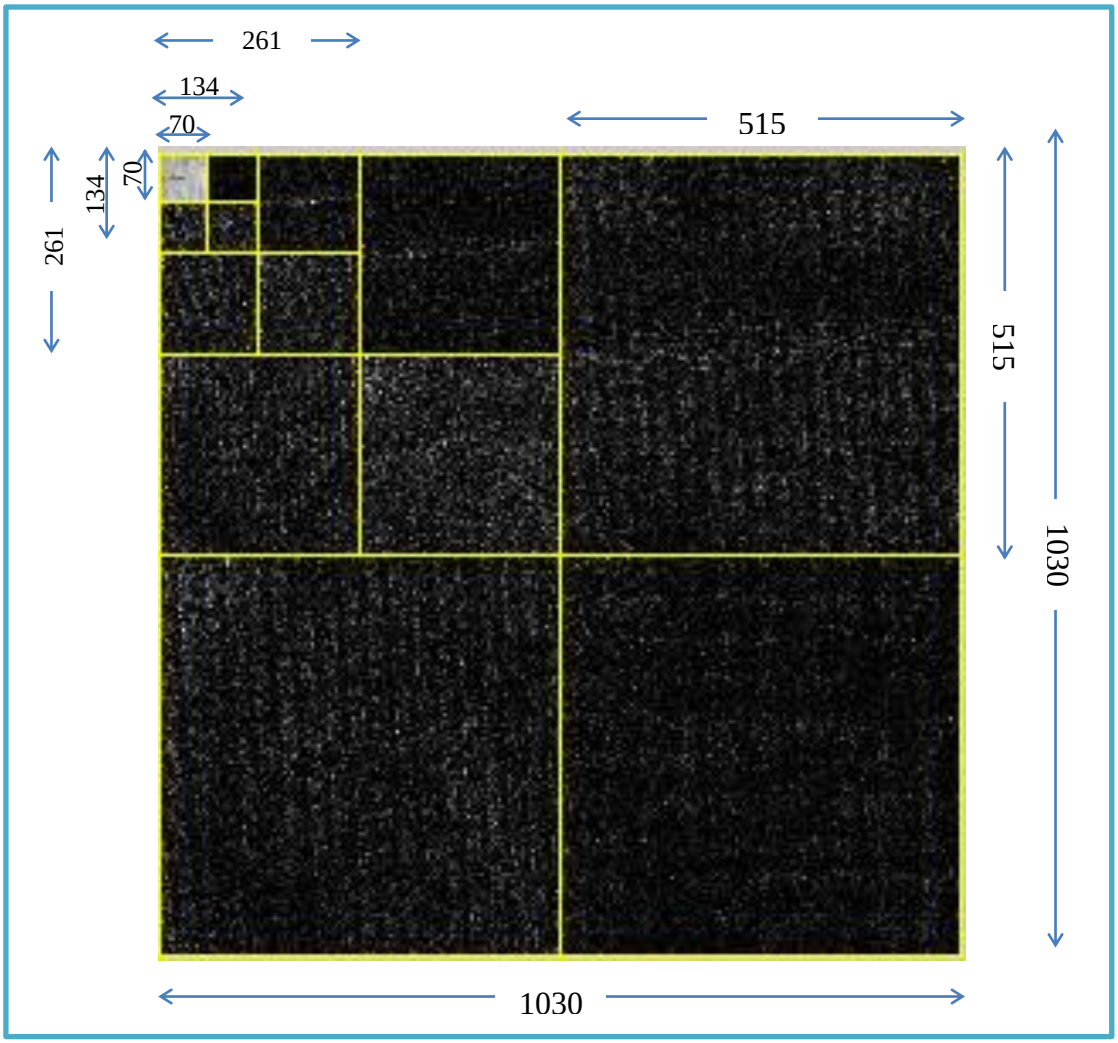
Hatalı kumaş resimleri M-dili ile yazılan programa RGB modunda yüklenmiştir, sonra RGB modundan gri seviye resimlerine dönüştürülmüştür (Şekil 4.1). Bu sayede resim veri matrisinin boyutu indirgenerek 2 boyutlu ayrık dalgacık dönüşümü (ADD) uygulanacak hale getirilmiştir. Örneğin; resim veri matrisi 256×256×3 olan bir RGB resmi gri seviye dönüştürülünce 256×256 matrisi haline gelmektedir.



Şekil 4.1. Faklı tip hata içeren kumaş resimlerinin gri-seviye dönüşmüş hali

Resim hakkında daha detaylı bilgi elde edebilmek için çoklu çözünürlük analizi (ADD) gerçekleştirilmiştir. Gri seviyeye dönüştürülmüş kumaş görüntüleri db-4 dalgacığı

kullanılarak 4. seviyeye kadar ayrıştırılmıştır (Şekil 3.19). Bu işlem bir önceki bölümde detaylı olarak izah edilmiştir. Uygulanan 4. seviye dönüşümden sonra alt bant resimleri (katsayıları) elde edilir. Bunlar; yaklaşım alt resmi, yatay alt resmi, dikey alt resmi ve çapraz alt resmidir (Şekil 4.2).



Şekil 4.2. Ayrışım gösterimi

Şekil 4.2'de görüldüğü üzere 1024×1024 boyutlarında kullandığımız kumaş görüntüsünün 1. seviye ayrışımından sonra elde edilen katsayılar; 515×515 yaklaşım, 515×515 yatay ayrıntı, 515×515 dikey ayrıntı ve 515×515 çapraz ayrıntı katsayılarıdır. 2. seviye için katsayılar; 261×261 yaklaşım, 261×261 yatay ayrıntı, 261×261 dikey ayrıntı ve 261×261 çapraz ayrıntı katsayılarıdır. 2. seviye için katsayılar; 261×261 yaklaşım, 261×261 yatay ayrıntı, 261×261 dikey ayrıntı ve 261×261 çapraz ayrıntı katsayılarıdır. 3. seviye için tüm katsayılar; 134×134 yaklaşım, 134×134 yatay ayrıntı, 134×134 dikey ayrıntı ve 134×134 çapraz ayrıntı katsayılarıdır. 4. seviye için tüm katsayılar; 70×70 yaklaşım, 70×70 yatay ayrıntı, 70×70 dikey ayrıntı ve 70×70 çapraz ayrıntı katsayılarıdır.

Tablo 4.1'de 4. seviye ayrışımından sonra elde edilen yaklaşım katsayılarından bir kesit verilmiştir.

Tablo 4.1. 4. seviyedeki yaklaşım katsayılarından bir kesit

Yaklaşım Katsayıları (1:10,1:5)				
2618,579	2618,892	2618,951	2618,267	2622,028
2618,415	2618,754	2618,799	2618,126	2621,876
2619,315	2619,740	2619,652	2619,007	2622,969
2617,927	2618,243	2618,270	2617,629	2621,322
2621,735	2622,182	2622,304	2621,451	2625,559
2608,039	2608,034	2607,784	2607,657	2610,560
2618,316	2618,919	2618,509	2617,805	2623,397
2644,257	2644,859	2643,580	2643,897	2647,575
2651,110	2651,476	2650,994	2650,991	2653,171
2631,266	2631,343	2629,656	2631,594	2629,038

CPU ve GPU'da Db4 dalgacığı kullanılarak gerçekleştirilen dalgacık dönüşümü işlem süreleri Tablo 4.2'de verilmiştir.

Tablo 4.2 Dalgacık dönüşümü işlem süreleri

Resimler	Dalgacık Dönüşümü		
	CPU Zamanı(sn)	GPU Zamanı(sn)	CPU/GPU
Resim 1	0,255	0,124	2,066
Resim 2	0,232	0,107	2,160
Resim 3	0,226	0,109	2,069
Resim 4	0,225	0,136	1,658
Resim 5	0,225	0,115	1,951
Resim 6	0,222	0,109	2,035
Resim 7	0,221	0,109	2,032
Resim 8	0,219	0,109	2,015
Resim 9	0,220	0,115	1,908
Resim 10	0,221	0,112	1,981
Resim 11	0,220	0,108	2,040
Resim 12	0,221	0,107	2,062
Resim 13	0,219	0,109	2,003
Resim 14	0,244	0,172	1,415
Resim 15	0,260	0,111	2,335
Resim 16	0,236	0,108	2,192
Resim 17	0,220	0,108	2,038
Resim 18	0,218	0,109	2,004
Resim 19	0,221	0,169	1,307
Resim 20	0,215	0,111	1,944
Resim 21	0,218	0,108	2,019
Resim 22	0,216	0,107	2,007
Resim 23	0,218	0,109	2,005
Resim 24	0,217	0,119	1,815
Resim 25	0,217	0,110	1,972

Tablo 4.2'de görülmekte olan kumaş resimlerinin dalgacık dönüşümü işlem verileri incelendiğinde her bir resim için GPU tabanlı uygulamada geçen süre ile CPU tabanlı uygulamada geçen süre oranı CPU/GPU her bir resim için yaklaşık 2 kattır. GPU tabanlı uygulamanın CPU tabanlı uygulamaya göre yaklaşık 2 kat daha hızlı olduğu görülmektedir.

Ayrıştırma işlemi sonucunda elde edilen 4. seviye yaklaşım katsayıları Y4 ve detay katsayıları D1 (1.seviye detay), D2 (2.seviye detay), D3 (3.seviye detay), D4 (4.seviye detay) kullanılarak öznitelik vektörü seçimi aşağıda açıklanan şekilde gerçekleştirilmiştir.

- Her alt-banttaki katsayıların medyanı

- Her alt-banttaki katsayıların ortalaması
- Her alt banttaki katsayıların standart sapması
- Her alt-banttaki katsayıların maksimumu
- Her alt-banttaki katsayıların minimumu

Yukarıdaki işlemlerle 1024×1024 giriş verisinden elde edilen dalgacık katsayıları (Y4 + A1 + A2 + A3 + A4 = 70×70 + 3×515×515 + 3×261×261 + 3×134×134 + 3×70×70) indirgenerek 70×70+60=4960 elemanlı bir veri kümesi elde edilmiştir. Böylece daha az veri girişi sağlanarak ağın eğitim süresi azaltılmıştır. Tablo 4.3'te db4 öznitelik vektörleri seçiminden bölüm verilmiştir.

Tablo 4.3. Öznitelik vektörleri seçimi (1. seviye katsayılarından)

İstatistiksel İşlemler	**Yatay**	**Dikey**	**Çapraz**
Medyan	0,0197438	0,0074890	-0,0008691
Ortalama	0,0111207	-0,0146949	-0,0034659
Standart Sapma	0,9063738	1,0208866	0,7753747
Maksimum	5,2575043	4,1995480	3,6737642
Minimum	-4,7429110	-9,1570558	-2,9142861

Tablo 4.3'te ise 1. seviye ayrıntı katsayılarının istatistiksel işlemler uygulanması ile elde edilen öznitelik vektörleri görülmektedir. Her bir kumaş görüntüsü için aynı şekilde öznitelik vektörleri seçimi hem CPU hem de GPU üzerinde gerçekleştirilmiştir. Her bir kumaş resmi için öznitelik vektörleri seçme işleminin CPU ve GPU üzerindeki işlem süreleri Tablo 4.4'te verilmiştir.

Tablo 4.4. Öznitelik seçimi işlem süreleri

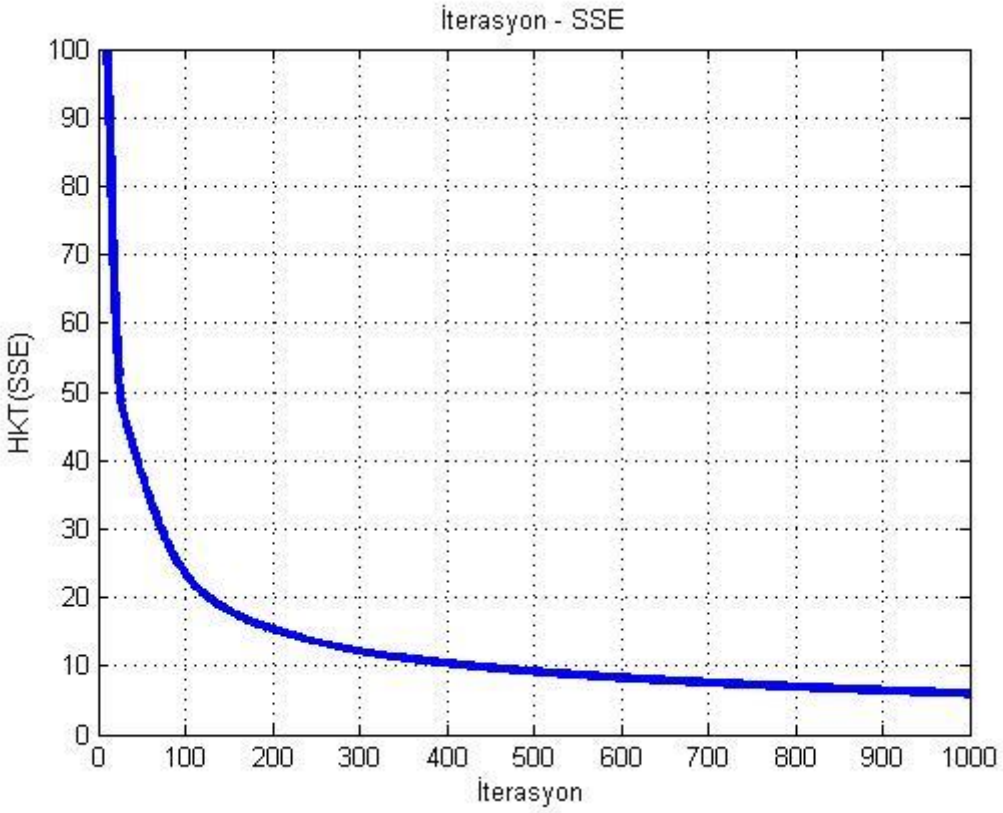
Resimler	Öznitelik Seçimi		
	GPU	CPU	CPU/GPU
Resim 1	0,0376	0,0320	0,8502
Resim 2	0,0344	0,0340	0,9897
Resim 3	0,0233	0,0327	1,4009
Resim 4	0,0332	0,0327	0,9856
Resim 5	0,0230	0,0337	1,4670
Resim 6	0,0224	0,0372	1,6559
Resim 7	0,0226	0,0325	1,4395
Resim 8	0,0228	0,0322	1,4131
Resim 9	0,0228	0,0340	1,4926
Resim 10	0,0225	0,0323	1,4313
Resim 11	0,0228	0,0336	1,4741
Resim 12	0,0230	0,0360	1,5665
Resim 13	0,0232	0,0325	1,3995
Resim 14	0,0222	0,0324	1,4581
Resim 15	0,0238	0,0333	1,4005
Resim 16	0,0228	0,0324	1,4214
Resim 17	0,0221	0,0354	1,6001
Resim 18	0,0226	0,0377	1,6707
Resim 19	0,0227	0,0322	1,4177
Resim 20	0,0233	0,0324	1,3877
Resim 21	0,0222	0,0322	1,4513
Resim 22	0,0231	0,0333	1,4426
Resim 23	0,0233	0,0392	1,6823
Resim 24	0,0231	0,0461	1,9962
Resim 25	0,0222	0,0900	4,0492

Tablo 4.4'te öznitelik vektörleri seçimine ilişkin işlem verileri incelendiğinde her bir resim için CPU tabanlı uygulamada geçen süre ile GPU tabanlı uygulamada geçen süreler oranı CPU/GPU yaklaşık 1,5 kat şeklindedir. Buna göre GPU tabanlı uygulamanın CPU tabanlı uygulamaya göre yaklaşık 1,5 kat daha hızlı olduğu görülmektedir. Her bir 1024×1024 ebadındaki kumaş resmi için elde edilen öznitelik vektörleri (4960 ×1) YSA'ya giriş olarak verilmeden önce normalizasyon işlemine tabi tutularak [-1,1] aralığına indirgenmiştir.

4.2 YSA sınıflandırması için CPU ve GPU performans Karşılaştırmaları

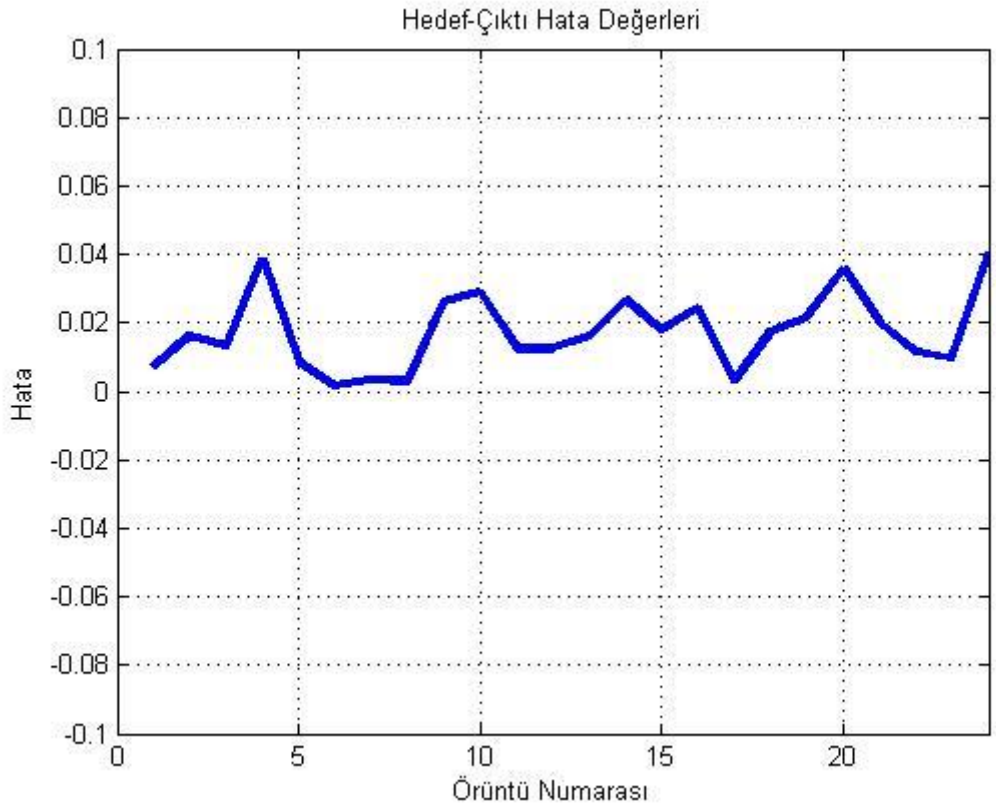
Uygulama da daha sonraki aşama ise elde edilen öznitelik vektörlerinin sınıflandırılma işlemidir. Sınıflandırma için giriş katman hücre sayısı: 4960, gizli katman

hücre sayısı:2000, çıkış katman hücre sayısı:3, performans fonksiyonu: Hata kareleri toplamı (SSE), gizli katman aktivasyon fonksiyonu: sigmoid, çıkış katman aktivasyon fonksiyonu: softmax olan geri yayılım algoritmasına sahip ileri beslemeli yapay sinir ağı oluşturulmuştur. Elde edilen öznitelik vektörleri oluşturulan bu YSA modeline giriş verisi olarak uygulanmış ve 1000 döngü sonucunda istenilen başarı düzeyine ulaşılmıştır. YSA'nın eğitim sırasında her bir döngü için hesaplanan hata kareleri toplamının değişimi aşağıdaki grafikte verilmiştir (Şekil 4.3).



Şekil 4.3. Döngü sayısına göre hata kareleri toplamı değişimi

Şekil 4.3'e bakıldığında 100 döngüden sonra hata kareleri toplamı 20 civarında iken 400 döngüden itibaren 10 civarında seyretmektedir. 1000 döngüden sonra hata kareleri toplamı 5'e kadar inmiştir. Şekil 4.4 eğitim işlemi tamamlandıktan sonra her bir örüntü için elde edilen ağ çıktı değerleri ile hedef çıktı değerleri arasındaki hataları göstermektedir. Buna göre hata değerleri 0 ile 0,04 arasında değişmektedir. Bu da yapılan sınıflandırmanın başarılı olduğunu göstermektedir.



Şekil 4.4. Her bir örüntü için hedef - çıktı arasındaki hata değerleri

YSA'da belirlediğimiz hedef çıktı değerleri; atkı hataları sınıflandırması için (1 0 0), çözgü hataları için (0 1 0) ve diğer tip hatalar için ise (0 0 1) şeklindedir. 3 hata tipi sınıflandırması için geliştirilen YSA'nın sınıflandırma başarı verileri Tablo 4.5'te verilmiştir.

Tablo 4.5 YSA sınıflandırma başarısı

Kumaş Hata Tipleri	Sınıflandırma Başarı Oranı (%)	SSE
Atkı Yönündeki Hatalar	96,9608	0,0199
Çözgü Yönündeki Hatalar	95,2650	0,0095
Diğer Tip Hatalar	94,5811	0,0386
Genel	95,6023	0,0680

Tablo 4.5'e bakıldığında kumaş hata tipleri için yaklaşık sınıflandırma başarı oranları; atkı yönündeki hatalar için % 97, çözgü yönündeki hatalar için % 95 ve diğer hata tipleri için ise %95'dir. Sistemin genel sınıflandırma başarı oranı ise %96 olarak hesaplanmıştır.

Tablo 4.6 YSA modeli ağ çıktı değerleri

Örüntü No	Hedef			Çıktı			Hata
1. Örüntü	1	0	0	0,9594	0,0013	0,0393	0,0406
2. Örüntü	1	0	0	0,9675	0,0051	0,0274	0,0325
3. Örüntü	1	0	0	0,9824	0,0130	0,0046	0,0176
4. Örüntü	1	0	0	0,9436	0,0356	0,0208	0,0564
5. Örüntü	1	0	0	0,9903	0,0041	0,0056	0,0097
6. Örüntü	1	0	0	0,9949	0,0008	0,0043	0,0051
7. Örüntü	1	0	0	0,9937	0,0000	0,0063	0,0063
8. Örüntü	1	0	0	0,9903	0,0000	0,0097	0,0097
9. Örüntü	1	0	0	0,9572	0,0114	0,0314	0,0428
10. Örüntü	1	0	0	0,8959	0,0174	0,0867	0,1041
11. Örüntü	1	0	0	0,9828	0,0120	0,0053	0,0172
12. Örüntü	1	0	0	0,9773	0,0017	0,0209	0,0227
13. Örüntü	0	1	0	0,0144	0,9664	0,0192	0,0336
14. Örüntü	0	1	0	0,0196	0,9452	0,0352	0,0548
15. Örüntü	0	1	0	0,0099	0,9381	0,0520	0,0619
16. Örüntü	0	1	0	0,0103	0,9609	0,0288	0,0391
17. Örüntü	0	0	1	0,0058	0,0017	0,9925	0,0075
18. Örüntü	0	0	1	0,0189	0,0323	0,9488	0,0512
19. Örüntü	0	0	1	0,0139	0,0048	0,9814	0,0186
20. Örüntü	0	0	1	0,0674	0,0226	0,9100	0,0900
21. Örüntü	0	0	1	0,0689	0,0025	0,9286	0,0714
22. Örüntü	0	0	1	0,0161	0,0222	0,9617	0,0383
23. Örüntü	0	0	1	0,0041	0,0065	0,9894	0,0106
24. Örüntü	0	0	1	0,1032	0,0169	0,8800	0,1200
25. Örüntü	0	0	1	0,0578	0,0223	0,9199	0,0801

Geliştirilen YSA modeli performans-işlem süreleri bakımından CPU ve GPU tabanlı olmak üzere eğitilmiş ve test edilmiştir. Eğitim ve test için geçen işlem süresi Tablo 4.7'de verilmiştir.

Tablo 4.7. YSA işlem süreleri

Örüntüler	YSA işlem süreleri		
	CPU	GPU	CPU/GPU
Eğitim	38,7082	3,9567	9,7830
Test	0,0600	0,0138	4,3347

Tablo 4.7'deki veriler incelendiğinde CPU'da eğitim için harcanan işlem süresi yaklaşık 39sn iken aynı işlem için GPU'da geçen işlem süresi yaklaşık 4sn kadardır. İşlem süreleri kıyaslandığında CPU'da eğitim aşaması için harcanan süre GPU'da eğitim aşaması için harcanan sürenin yaklaşık 10 katıdır. GPU, CPU göre 10 kat daha hızlı çalışmaktadır.

Benzer kıyaslama CPU ve GPU'daki test aşaması için yapıldığında CPU'da harcanan sürenin GPU'da harcanan süreye göre yaklaşık 4,5 kat daha fazla olduğu görülmektedir. Tablo 4.8'de uygulamaya ait genel veriler verilmiştir.

Tablo 4.8. Uygulama aşamalarına ait işlem süreleri

Uygulama	CPU	GPU	CPU/GPU
Öznitelik Çıkarımı	2,1948	1,0777	2,0365
Öznitelik Seçimi	0,9119	0,6070	1,5023
YSA İşlem Süresi	38,7681	3,9705	9,7640
Toplam Cevap Süresi	63,5508	10,2598	6,1942

Tablo 4.8'de uygulama aşamaları için harcanan işlem süreleri incelendiğinde CPU'da geçen toplam işlem cevap süresinin GPU'da geçen toplam işlem cevap süresine oranı CPU/GPU yaklaşık 6'dır. Sonuç olarak GPU 6 kat daha hızlı çalışmaktadır.

5. SONUÇ

Bu tez çalışmasında otomatik olarak kumaş hatalarının tespiti ve sınıflandırılması için M-dilinde (Matlab programı) program yazılmıştır. Bu program örüntü tanıma işlem basamaklarını içermektedir. Örüntü tanıma işlem basamakları ön işleme, öz nitelik çıkarma/seçme ve sınıflandırma aşamalarından oluşmaktadır.

İlk olarak RGB modundaki kumaş hata resimleri gri-seviye formatına dönüştürülmüştür. Daha sonra gri seviye resimlerine ADD uygulanarak görüntü hakkında önemli bilgi içeren dalgacık katsayıları elde edilmiştir. Dalgacık katsayılarının hesaplanmasında 4. seviye Daubechies dalgacıkları kullanılmıştır. YSA uygulanacak giriş veri boyutlarını azaltmak için ADD'den elde edilen katsayılar istatistiksel işlemlere tabi tutulmuştur. Sınıflandırma için ileri beslemeli ve geri yayılımlı YSA modeli kullanılmıştır. Kumaş hata tespiti ve sınıflandırması için geliştirilen uygulama hem CPU ve GPU'da çalıştırılarak hız bakımından performans değerlendirmesi yapılmıştır. Sonuç olarak

- 3 hata tipi sınıflandırması için geliştirilen YSA'nın sınıflandırma başarı oranları atkı yönündeki hatalar için % 97, çözgü yönündeki hatalar için %95, diğer tip hatalar için ise %95 ve genel olarak sınıflandırma başarısı %96 olarak elde edilmiştir.
- Çalışma hızı performansı açısından GPU yaklaşık olarak öznitelik çıkarımı için 2, öznitelik seçimi için 1,5, YSA sınıflandırması için 10 ve tüm uygulamanın genel işlem süresi için ise 6 kat daha hızlı olduğu görülmüştür.
- GPU tabanlı geliştirilen uygulama gerçek zamanlı çevrimiçi hata kontrol sistemleri için CPU tabanlı uygulamalara göre daha avantajlıdır.

KAYNAKLAR

Alimohamadi, H., Ahmadyfard, A., Shojaee, E., 2009. Defect Detection in Textiles Using Morphological Analysis of Optimal Gabor Wavelet Filter Response. Bangkok, Thailand. International Conference on Computer and Automation Engineering, 8-10 March 2009, Semnan Technology. Park, Semnan, Iran. s. 26-30.

Basu, M., Lin, Z. Y., 1992. Multi-scale Modeling of Texture. The Hague, Holland. Proceedings International Conference on Pattern Recognition: Image, Speech and Signal Analysis, 30 Aug-3 Sep 1992, City University of New York, New York. s. 421-424.

Baştürk, A., Ketencioglu, H., Yugnak, H., Yüksel, M. E., 2007. Inspection of Defects in Fabrics Using Gabor Wavelets and Principle Component Analysis. Sharjah, United Arab Emirates. 9th International Symposium on Signal Processing and Its Applications, 12-15 Feb. 2007, Erciyes University, Kayseri, Turkey. s. 1-4.

Baştürk, A., Ketencioglu, H., Yugnak, Z., Yildiz, C., Yuksel, M. E., 2005. Detection of Local Defects in Fabrics Using Gabor Filter Banks. Proceedings of the IEEE 13th. Signal Processing and Communications Applications Conference, 16-18 May 2005, Erciyes University., Kayseri, Turkey. s. 17-20.

Bodnarova, A., Bennamoun, M., Kubik, K. K., 1998. Defect Defection in Textile Materials Based on Aspects on The HVS. San Diego, USA. Proceedings of IEEE Intelligence Conference Systems, Man and Cybernetics, 11-14 Oct 1998, Queensland University, Brisbane, Australia. s. 4423-4428.

Bodnarova, A., Bennamoun, M., Latham, S., 2002. Optimal Gabor Filters for Textile Flaw Detection. *Pattern Recognition*, 35 : 2973-2991.

Bodnarova, A., Bennamoun, M., Latham, S. J., 2000. A Constrained Minimisation Approach to Optimise Gabor Filters for Detecting Flaws in Woven Textiles. Istanbul, Turkey. Proceedings of IEEE International Conference on Acoustics on Speech and Signal Processing, 05-09 Jun 2000, Queensland University., Brisbane, Australia. s. 3606-3609.

Bodnarova, A., Williams, J., Bennamoun, M., Kubik, K., 1997. Optimal Textural Features for Flaw Detection in Textile Materials. Brisbane, Australia. Proceedings of IEEE TENCON'97 Conference, 4-4 Dec. 1997, Queensland University, Brisbane, Australia. s. 307-310.

Bu, H. G., Wang, J., Huang, X., 2009. Fabric defect detection based on multiple fractal features and support vector data description. *Engineering Applications of Artificial Intelligence*, 22(2) : 224-235.

Busch, C., 1997. Wavelet Based Texture Segmentation of Multi-Modal Tomographic Images. *Computer and Graphics*, 21(3) : 347-358.

Campbell, J. G., Hashim, A. A., Murtagh, F. D., 1997. Flaw Detection in Woven Textiles using Space-dependent Fourier Transform. Ulster, UK. INFM-97-004, 16 May 1997, University of Ulster. UK. s. 1-8

Campbell, J. G., Hashim, A., McGinnity, T. M., Lunney, T. F., 1997. Flaw Detection in Woven Textiles by Neural Network. INFM-97-002, 10 May 1997, University of Ulster. UK. s. 1-8

Casterllini, C., Francin, F., Longobardi, G., Tribilli, B., 1996. On-line Textile Quality Control Using Optical Fourier Transforms. *Optics and Lasers in Engineering,* 24 : 19-32.

Castilho, H. P., Goncalves, P. J., Pinto, J. R., Serafim, A. L., 2007. Image Analysis and Recognition. Proceedings Book Series: Lecture Notes in Computer Science, s. 1297-1307.

Chan, C. H., Pang, G. K., 2000. Fabric Defect Detection by Fourier Analysis. *IEEE Transactions on Industry Applications,* 36 : 1267-1276.

Chan, H. Y., Raju, C., Sari-Sarraf, H., Hequet, E. F., 2005. A General Approach to Defect Detection in Textured Materials Using a Wavelet Domain Model and Level Sets. Texas, USA. Proceedings of The International Society for Optics and Photonics on Wavelet Applications in Industrial Processing III, November 2005, Texas Tech. University., USA s. 102-107.

Chen, C. H., Pau, L. F., Wang, P. S., 1993. Handbook of Pattern Recognition and Computer Vision. World Scientific Publishing Co. ISBN: 10 981-4273-38-4. 967s

Chen, J., Jain, A. 1988. A structural Approach to Identify Defects in Textured Images. Proceedings of IEEE International Conference on Systems, Man and Cybernetics, 8-12 Aug 1988, Suzhou, China s. 29-32.

Chen, S., Feng, J., Zou, L., 2010. Study of Fabric Defects Detection Through Gabor Filter Based on Scale Transformation. Zhejiang, China. International Conference on Image Analysis and Signal Processing, 9-11 April 2010, Jiangsu Polytech University., Changzhou, China. s. 97-99.

Chetverikov, D., 2000. Structural Defects: General Approach and Application to Textile Inspection. Barcelona, Spain. Proceedings of IEEE 15th International Conference Pattern Recognition, 03-07 Sep 2000, Budapest, Hungary. s. 521-524.

Chetverikov, D., Hanbury, A., 2002. Finding Defects in Texture Using Regularity and Local Orientation. *Pattern Recognition,* 35 : 2165-2180.

Chiu, S. H., Chou, S., Liaw, J. J., Wen, C. Y., 2002. Textural defect segmentation using a Fourier-Domain Maximum Likelihood Estimation Method. *Textile Research Journal,* 72 : 253-258.

Chung-Feng, J., Kuo, C.-J. L., 2003. A Back-Propagation Neural Network for Recognizing Fabric Defects. *Textile Research Journal,* 73(2) : 147-151.

Ciamberlini, C., Francini, F., Longobardi, G., Pog, P., 1996. Weaving Defect Detection by Fourier Imaging. *Vision System Application,* 2786 : 9-18.

Cohen, F. S., Fan, Z. G., Attali, S., 1991. Automated Inspection of Textile Fabric Using Textural Models. *IEEE Transactions Pattern Analysis and Machine Intelligence,* 13(8) : 803-808.

Conci, A., Proença, C. B., 1998. A Fractal Image Analysis System for Fabric Inspection Based on A Box-Counting Method. *Computer Networks and ISDN Systems,* 30 : 1887-1895.

Conners, R. W., McMillin, C. W., Lin , K., Vasque, R. E., 1983. Identifying and Locating Surface Defects in Wood: Part of An Automated Lumber Processing System. *IEEE Transactions on Pattern Recognition and Machine Intelligence,* 5 : 573-583.

Daubechies, I., 1990. The Wavelet Transform Time-Frequency Localization and Signal Analysis. *IEEE Transactions on Information Theory,* 36(5) : 961-1005.

Daubechies, I., 1992. Ten Lectures on Wavelets. *Society for Industrial and Applied Mathematics*. Philadelphia.

Daugman, J. G., 1980. Two-Dimensional Spectral Analysis of Cortical Receptive Field Profiles. *Vision Research,* 20(10) : 847-856.

Duda, R. O., Hart, P. E., Stork, D. G., 2001. Pattern Classification. Wiley-Interscience, John Wiley Sons Inc. 654s.

Elman, J. L., 1990. Finding Structure in Time. *Cognitive Science,* 14 : 179-211.

Elmas, Ç., 2007. Yapa Zeka Uygulamaları. Ankara. Seçkin Yayıncılık Sanayi ve Ticaret A.Ş. ISBN: 978-975-02-0614-6, Türkiye, 423s.

Escofet, J., Millan, S., Abril, H., Torrecilla, E., 1998. Inspection of Fabric Resistance to Abrasion by Fourier Analysis. Proceedings of The International Society for Optics and Photonics in Computing, 22 May 1998, s. 207-210.

Escofet, J., Navarro, R., Millan, M. S., 1998. Detection of Local Defects in Textile Webs Using Gabor Filters. *Optical Engineering*, 37(8) : 2297-2307.

Fausett, L., 1994. Fundamentals of Neural Networks. Englewood, NJ: Prentice Hall.

Feldman, J. A., Ballard, D. H., 1982. Connectionist Models and Their Properties. *Cognitive Science,* 6 : 205 254.

Garland, M., Le Grand, S., Nickolls, J., Anderson, J., Hardwick, J., Morton, S., Phillips, E., Zhang, Y. and Volkov, V., 2008. Parallel Computing Experiences with CUDA. *Micro, IEEE*, 284 : 13-27.

Grap, A., 1995a. An Introduction To Wavelets. IEEE Computational Science and Engineering, 2(2).

Guan, S., 2010. Fabric Defect Detection Based on Fusion Technology of Multiple Algorithms. International Conference on Signal Processing Systems, s. 553-557.

Guan, S., Shi, X., 2008. Fabric Defect Detection Based on Wavelet Decomposition With One Resolution Level. IEEE International Symposium on Information Science and Engineering, s. 281-285.

Guan, S., Yuan, J., Ma, K., 2011. Fabric Defect Detection Based on Wavelet Reconstruction. International Conference on Multimedia Technology, s. 3520-3523.

Gümüş, İ., 2003. EKG Sinyallerinin Wavelet Analizi. Lisans Tezi. Uludağ Üniversitesi Mühendislik Mimarlık Fakültesi Elektronik Mühendisliği Bölümü. Bursa. 158s.

Han, Y., Shi, P., 2007. An Adaptive Level-selecting Wavelet Transform for Texture Defect Detection. *Image and Vision Computing*, 25 : 1239-1248.

Haralick, R. M., Shanmugam, K., Dinstein, I., 1973. Textural Features for Image Classification. *IEEE Transactions Systems, Man and Cybernetics*, 3(6) : 610-621.

Hoffer, L. M., Francini, F., Tiribilli, B., Long, G., 1996. Neural Network for The Optical Recognition of Defects in Cloth. *Optical Engineering*, 35(11) : 3183-3190.

Hou, Z., Parker, J. M., 2005. Texture Defect Detection Using Support Vector Machines with Adaptive Gabor Wavelet Features. Seventh IEEE Workshops on Applications of Computer Vision. s. 275-280.

Hung, C. C., Chen, I. C., 2001. Neural-Fuzzy Classification for Fabric Defects. *Texture Research Journal*, 71(3) : 220-224.

Jacket, 2012. Ocean Carbon-Cycle Model Intercomparison Project, IGBP, Global, Analysis, Interpretation And Modeling Task Force, http:/www.Ipsl.jussieu.-fr/OCMIP.

Jain, A. K., Farrokhnia, F., 1991. Unsupervised Texture Segmentation Using Gabor Filters. *Pattern Recognition*, 24(2) : 1167-1186.

Jain, A. K., Duin, R. P., Mao, J., 2000. Statistical Pattern Recognition: A Review. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(1) : 4-37.

Jang, H., Park, A., Jung, K., 2008. Neural Network Implementation Using CUDA and OpenMP. *Digital Image Computing: Techniques and Applications*, s. 155-161.

Jayashree, V., Subbaraman, S., 2011. DCSFPSS Assisted Morphological Approach for Grey Twill Fabric Defect Detection and Defect Area Measurement for Fabric Grading. International Symposium Electronic System Design, s. 290-295.

Jiang, H., Dong, M., Li, W., 2009. Detection of Fabric Defect Based on Optimal Tree Structure of Wavelet Decomposition. International Symposium on Intelligent Ubiquitous Computing and Education, s. 210-213.

Jordan, M. I., 1986. Attractor Dynamics and Parallelism in a Connectionist Sequential Machine. Proceedings of the Eighth Annual Conference of the Cognitive Science Society, s. 531-546.

Kim, S., Lee, M., Woo, K., 1999. Wavelet Analysis to Defects Detection in Weaving Processes. Proceedings of the IEEE International Symposium on Industrial Electronics, 3, s. 1406–1409.

Kumar, A., 2003. Neural Network Based Detection of local Textile Defects. *Pattern Recognition,* 36 : 1645-1659.

Kumar, A., Pang, G., 2000. Fabric Defect Segmentation Using Multichannel Blob Detectors. *Optical Engineering,* 39(12) : 3176-3190.

Kumar, A., Pang, G., 2002. Defect Detection in Textured Materials Using Gabor Filters. *IEEE Transactions Industry Applications,* 38 : 425–440.

Kumar, A., Shen, H. C., 2002. Texture Inspection for Defects Using Neural Networks and Support Vector Machines. Proceedings International Conference Image Processing, Rochester, New York. s. 353-356.

Kuo, C. F., Lee, C., 2003. A Back-Propagation Neural Network for Recognizing Fabric Defects. *Textile Research Journal,* 73(2) : 147-151.

Latif-Amet, A., Ertuzun, A., Ercil, A., 1998. Texture Defect Detection Using Subband Domain Co-occurrence Matrices. IEEE Southwest Symposium on Image Analysis and Interpretation, s. 205-210.

Latif-Amet, L., Ertuzun, A., Ercil, A., 2000. An Efficient Method for Texture Defect Detection: Subband Domain Co-occurrence Matrices. *Image and Vision Computing,* 18 : 543–553.

Levy, M. N., Martin, P. J., 1979. Neural control of the heart, In: Berne R.M. ed.: Handbook of Physiology.

Liang, Z., Xu, B., Chi, Z., Feng, D., 2012. Intelligent Characterization and Evaluation of Yarn Surface Appearance Using Saliency Map Analysis, Wavelet Transform and Fuzzy ARTMAP Neural Network. *Expert Systems with Applications,* 9(4) : 4201-4212.

Liu, G. S., Qu, P. G., 2008. Inspection of Fabric Defects Based on Wavelet Analysis and BP Neural Network. Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition, Hong Kong. s. 232-236.

Liu, H., Han, J., 2006. Defect Detection in Textiles Using Optimal Gabor Wavelet Filter. IEEE Proceedings The 6th World Congress on Intelligent Control and Automation, Dalian, China. s. 10005–10007.

Liu, J., Zuo, B., Zeng, X., Vroman, P., Rabenaso, B., 2010. Nonwoven Uniformity Identification Using Wavelet Texture Analysis and LVQ Neural Network. *Expert Systems with Applications,* 37(3) : 2241-2246.

Liu, J., Zuo, B., Zeng, X., Vroman, P., Rabenaso, B., 2011a. Wavelet Energy Signatures and Robust Bayesian Neural Network for Visual Quality Recognition of Nonwovens. *Expert Systems with Applications,* 38(7) : 8497-8508.

Liu, X., Wen, X., Su, X., Choi, K. F., 2008. Slub Extraction in Woven Fabric Images Using Gabor Filters. *Textile Research Journal*, 78(4) : 320–325.

Mahajan, P. M., Kolhe, S. R., Patil, P. M., 2009. A Review of Automatic Fabric Defect Detection Techniques. *Advances in Computational Research*, 1(2) : 18-29.

Mak, K. L., Peng, P., Lau, H. Y., 2005. Optimal Morphological Filter Design for Fabric Defect Detection. IEEE International Conference on Industrial Technology, Hong Kong, China. s. 469-474.

Mak, K. L., Peng, P., Yiu, K. F., 2009. Fabric Defect Detection Using Morphological Filters. *Image and Vision Computing*, 27(10) : 1585–1592.

Mak, K., Peng, P., 2008. An Automated Inspection System for Textile Fabrics Based on Gabor Filters. Robotics and Computer-Integrated Manufacturing, 24 : 359-369.

Mak, K., Tian, X. W., 2010. Iterative Tensor Tracking Using GPU for Textile Fabric Defect Detection. International Conference Green Circuits and Systems. Hong Kong, China.

Mallat, S. G., 1989. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 11(7) : 674-693.

Mattoccia, S., Viti, M., Ries, F., 2011. Near Real-time Fast Bilateral Stereo on the GPU. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Bologna. s. 136-143.

McCulloh, W., Pitts, W., 1943. A Logical Calculus of The Ideas Immanent Activity. *Bulletin of Mathematical Biophysics*, 5 : 115-133.

Membarth, R., Dutta, H., Hanning, F., Teich, J., 2011. Efficient Mapping Of Streaming Applications for Image Processing On Graphics Cards. Transactions On Hipeac, 5(3).

Minsky, M. L., Papert, S. A., 1969. Perceptron, Expanded Edition. Cambridge: MIT Press, London, UK.

Misiti, M., Misiti, Y., Oppenheim, G., J-M., P., 1996. Wavelet Toolbox User's Guide. MathWorks.

Ngana, H., Panga, G. K., Yung, N., 2011. Automated Fabric Defect Detection-A review. *Image and Vision Computing*, 29 : 442-458.

Nvidia. Cuda C Programming Guide, NVIDIA Corp., URL: erişim tarihi:19/05/2012 http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf

Nvidia. Nvidia Developer Zone, Toolkit, URL: erişim tarihi:09/08/2012 http://developer.nvidia.com/cuda-toolkit-40

Nvidia Corp., Nvidia Geforce GTX 480/470/465 Datasheet URL: erişim tarihi: 01/09/2012 http://www.geforce.com/active/en_us/en_us/pdf/gtx-480-470-web-datasheet.pdf

Ogata, N., Fukuma, S., Nishikado, H., Shirosaki, A., Takagi, S., Sakurai, T., 2005. An Accurate Inspection of PDP-mesh Cloth Using Gabor Filter. Proceedings IEEE International Symposium Intelligent Signal Processing Communication Systems, Hong Kong. s. 65-68.

Owens, J., 2008. GPU Computing. Proceedings of the IEEE, 98, s. 879-899.

Özdemir, S., Ercil, A., 1996. Markov Random Fields and Karhunen–Loeve Transform for Defect Inspection of Textile Products. Proceedings IEEE Conference Emerging Technologies and Factory Automation, 2, s. 697-703.

Özdemir, S., Baykut, A., Meylani, R., Ercil, A., Ertuzini, A., 1998. Comparative Evaluation of Texture Analysis Algorithms for Defect Inspection of Textile Products. Proceedings of the IEEE 14th International Conference on Pattern Recognition, 2, s. 1738-1740.

Parker, D. B., 1985. Learning Logic. Tech. Rep. Nos. TR-47, 33.

Pearlmutter, B. A., 1990. Dynamic Recurrent Neural Networks. Tech. Rep. Nos., s. 17-50.

Pentland, A. P., 1984. Fractal-Based Description of Natural Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6) : 661-674.

Percival, D. B., Walden, A. T., 2002. Wavelet Methods for Time Series Analysis. Cambridge University Press.

Ravandi, S. A., Toriumi, K., 1995. Fourier Transform Analysis of Plain Weave Fabric Appearance. *Textile Research Journal*, 65(11) : 676–683.

Rohrmus, D., 2000. Invariant Web Defect Detection and Classification System. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2, s. 794-795.

Rösler, R. N., 1992. Defect Detection in Fabrics by Image Processing. *Mellind Textilberichte*, 73 : 292.

Sari-Sarraf, H., Goddard, J. S., 1999. Vision System for On-loom Fabric Inspection. *IEEE Transactions on Industry Applications*, 35(6) : 1252-1259.

Sengottuvelan, P. A., Wahi, A., Shanmugam, A., 2008. Automatic Fault Analysis of Textile Fabric Using Imaging Systems. *Research Journal of Applied Science*, 3 (1), 26-31.

Serdaroğlu, A., Ertüzün , A., A.Erçil, Erçil, A., 2006. Defect Detection in Textile Fabric Images Using Wavelet Transforms and Independent Component Analysis. *Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications*, 16 : 61-64.

Shi, M., Jiang, S., Wang, H., Xu, B., 2009. A Simplified Pulse-coupled Neural network for Adaptive Segmentation of Fabric Defects. *Machine Vision and Applications,* 20(2) : 131-138.

Shu, Y., Tan, Z., 2004. Fabri Defects Automatic Detection Using Gabor Filters. IEEE Proceedings The 5th World Congress on Intelligent Control and Automation, Hangzhou, China, s. 3378–3380.

Steinkrau, D., Simard, P. Y., Buck, I., 2005. Using GPUs for Machine Learning Algorithms. Proceedings of the Eighth International Conference on Document Analysis and Recognition, s. 1115-1119.

Su, T. L., Chen, H. W., Hong, G. B., Ma, C. -M., 2010. Automatic Inspection System for Defects Classification of Stretch Knitted Fabrics. International Conference on Wavelet Analysis and Pattern Recognition, s. 125-129.

Torrence, C., Compo, P. C., 1998. A Practical Guide To Wavelet Analysis. *Bulletin Of American Meteorological Society,* 79(1) : 61-78.

Tsai, D. M., Huang, T. Y., 2003. Automated Surface Inspection for Statistical Textures. *Image and Vision Computing,* 21 : 307-323.

Tsai, D. M., Hsieh, C. Y., 1999. Automated Surface Inspection for Directional Textures. *Image and Vision Computing,* 18 : 49-62.

Tsai, D. M., Lin, C. P., 2002. Fast Defect Detection in Textured Surfaces Using 1D Gabor Filters. *The International Journal of Advanced Manufacturing Technology,* 20 : 664-675.

Tsai, D. M., Wu, S. K., 2000. Automated Surface Inspection Using Gabor Filters. *The International Journal of Advanced Manufacturing Technology, 16*, s. 474-482.

Tüfekci, Z., Gowdy, J. N., 2000. Feature Extraction Using Discrete Wavelet Transform for Speech Recognition. Proceedings of the IEEE Southeast, s. 116-123.

Türkoğlu, İ., 2002. Durağan Olmayan İşaretler için Zaman-Frekans Entropilerine Dayalı Akılllı Örüntü Tanıma. Doktora Tezi. Fırat Üniversitesi Fen Bilimleri Enstitüsü. Elazığ. 131s.

Uçan, O. N., 2003. İşaret ve Görüntü İşlemede Yeni Yaklaşımlar. İ.Ü. Mühendislik Fakültesi Yayınları No:4408. İstanbul. 213s.

Unser, M., 1995. Texture Classification and Segmentation Using Wavelet Frames. *IEEE Trans. Image Processing,* 4 : 549-1560.

Vilnrotter, F., Nevatia, R., Price, K., 1986. Structural Analysis of Natural Textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 8 : 76-89.

Wiklund, U., Akay, M., Niklasson, U., 1997. Short-term Analysis of Heart-rate Variability by Adapted Wavelet Transforms. *IEEE Engineering Med. Biol. Mag.*, s. 113-138.

Wood, E. J., 1990. Applying Fourier and Associated Transforms to Pattern Characterization in Textiles. *Textile Research Journal*, 60 : 212-220.

Yang, X., Pang, G., Yung, N., 2005. Robust Fabric Defect Detection and Classification Using Multiple Adaptive Wavelets. IEE Proceedings Vision Image Signal Process, 1526, s. 715-723.

Yang, Z., Zhu, Y., Yong, P., 2008. Parallel Image Processing Based on CUDA. International Conference on Computer Science and Software Engineering, s. 198-201.

Yin, Y., Lu, W. B., Zhang, K., Jing, L., 2009. Textile Flaw Detection and Classification by Wavelet Reconstruction and BP Neural Network. *WRI Global Congress on Intelligent Systems*, 4 : 167-171.

Yuen, C. W., Wong, W. K., Qian, S. Q., Chan, L. K., Fung, E., 2009. A Hybrid Model Using Genetic Algorithm and Neural Network for Classifying Garment Defects. *Expert Systems With Applications,* 36 : 2037-2047.

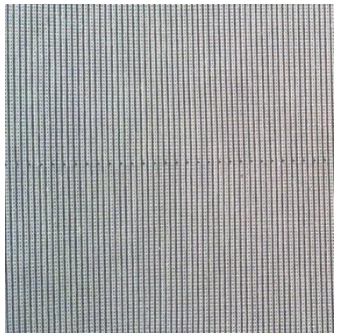
Zhang, W., Zhao, Q., Liao, L., 2010. Development of a Real-time Machine Vision System for Detecting Defeats of Cord Fabrics. International Conference on Computer Application and System Modeling, 12, s. 539-543.

Zhang, Y. F., Bresee, R. R., 1995. Fabric Defect Detection and Classification Using Image Analysis. *Texture Research Journal*, 65 : 1-9.

Zhang, Y., Lu, Z., Li, J., 2010. Fabric Defect Detection and Classification Using Gabor Filters and Gaussian Mixture Model, Asian Conference Computer Vision, Part II, Lecture Notes in Computer Science, s. 635-644.

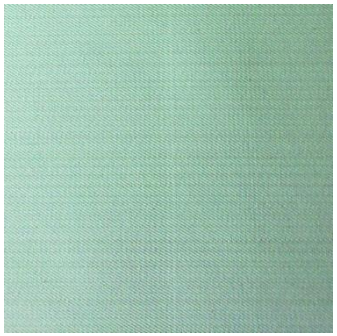
Zhao, D. X., Wang, H., Zhu, J. L., Li, J. L., 2008. Research on a New Fabric Defect Identification Method. International Conference on Computer Science and Software Engineering, 2, s. 814-817.

EKLER

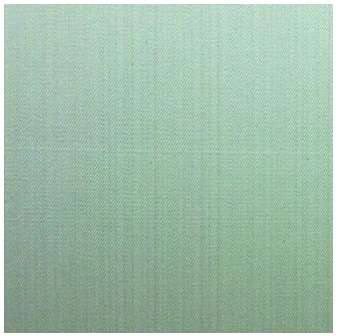


Resim 1-Atkı Kaçığı

Resim 2-Atkı Kaçığı



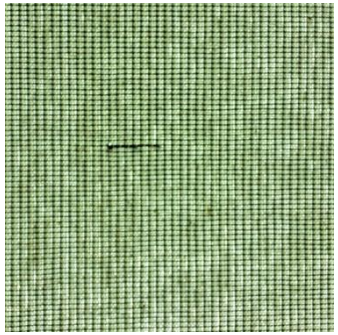
Resim 3-Atkı Kaçığı



Resim 4-Atkı Kaçığı

Resim 5-Atkı Kaçığı

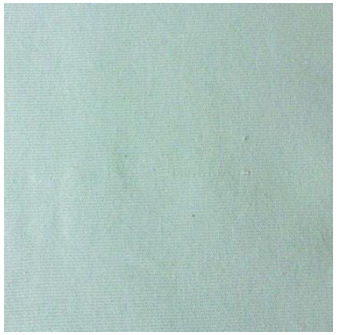
Resim 6-Atkı Yığılması



Resim 7-Atkı Yığılması

Resim 8-Çift Atkı ve Yığılma

Resim 9-Çözgü Kaçık



Resim 10-Çözgü Kaçık ve Düğüm

Resim 11-Çözgü Kaçık

Resim 12-Çözgü Kaçık

Resim 13-Düğüm

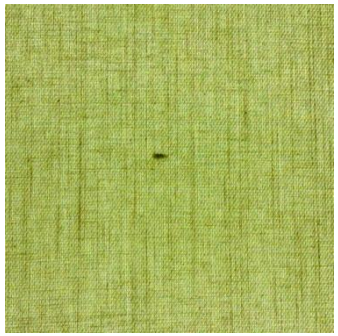
Resim 14-Kalın Atkı

Resim 15-Kalın Atkı

Resim 16-Şantuk

Resim 17-Yabancı Madde

Resim 18-Yabancı Madde



Resim 19-Yabancı Madde

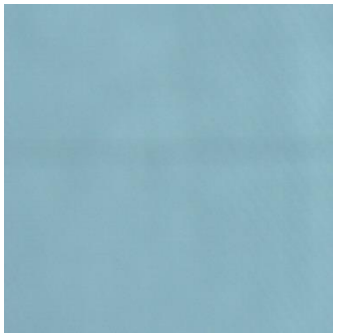
Resim 20-Yabancı Madde

Resim 21-Yabancı Madde

Resim 22-Yağ Lekesi

Resim 23-Yağ Lekesi

Resim 24-Yığılma



Resim 25-Atkı Bandı

Ek-1. Dijital kamera ile KİPAŞ firmasından alınan hatalı kumaş örnekleri

ÖZGEÇMİŞ
Eylül 2012

Kişisel Bilgiler

Adı soyadı	**:Hasan Güler**
Adres	**:**Kilis 7 Aralık Üniversitesi Merkez/KİLİS
Doğum Tarihi	**:**05.11.1982
Medeni Hal	**:**Evli
Telefonla	**:** 05054325965
E-Posta	**:**hasanguler@kilis.edu.tr
Askerlik Durumu	**:**Tamamlandı(Yedek Subay olarak)

İş Deneyimleri

2004–2006	: GKV Özel Okulları - Gaziantep Bilgisayar Öğretmeni
2006-2010	: Gaziosmanpaşa İlköğretim Okulu Bilişim Öğretmeni (Formatör)

Eğitim Bilgileri

Lise	**:**Nizip Endüstri Meslek Lisesi 1996-1999
Üniversite	**:**Orta Doğu Teknik Üniversitesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi 1999-2004
Yabancı Dil	**:**İngilizce KPDS: 72 puan

Bilgisayar Deneyimi:

Microsoft Office Uygulamaları(MS Word,Excel,Access,Powerpoint), Adobe CS3 Uygulamaları(Dreamweaver, Photoshop, Flash and Action Script 3,0, Illustrator, Fireworks), C# .NET, PHP, ASP, HTML, MySQL, Visual Basic Programlama Dili, Javascript, Windows XP, Windows 2003 Server ve Networking Systems.

Kurs ve Seminerler:

IT Essentials I: PC Hardware and Software Cisco Networking Academy	01/08/2005	05/08/2005
İntel Gelecek İçin Eğitim Kursu	30/10/2006	10/11/2006
Yeni İlköğ.Prog.Tanıtım Semineri(Bilgisayar)	13/01/2007	14/01/2007
Web Tabanlı İçerik Geliştirme Kursu (I. Kademe)	02/02/2009	13/02/2009
Web Tabanlı İçerik Geliştirme Kursu (II. Kademe)	18/05/2009	30/05/2009

Hobiler

Futbol, Voleybol, Yüzme, Dağ Sporları, Kara Kalem, Teknoloji