

- מערכת זיהוי כתב יד בעברית
 - ארכיטקטורה כללית
 - מבנה תיקיות
 - שלב 1: הקמת תשתית בסיסית
 - 1.1 Backend - Express + Firebase
 - 1.2 Frontend - Next.js
 - 1.3 Firebase Setup
 - (CRITICAL) שלב 2: שיפור וחיידוד תמונות
 - 2.1 Image Enhancement Pipeline
 - 2.2 פילטרים זמינים
 - 2.3 ממשק שיפור תמונות (Real-time)
 - 2.4 תכונות ממשק המשתמש
 - 2.4.1 קומפוננטים נדרשים
 - 2.5 Presets מוכנים
 - 2.6 WebSocket לתצוגה בזמן אמת
 - 2.7 API Endpoints לשיפור תמונות
 - 2.8 שמירת הגדרות לכל תמונה
 - 2.9 Python Enhancement Service - מימוש טכני
 - 2.10 אופטימיזציה לביצועים
 - שלב 3: עיבוד תמונות וסגמנטציה
 - 3.1 Image Processor
 - 3.2 Firestore-מבנה נתונים ב
 - (Annotation Interface) שלב 4: ממשק תיוג
 - 4.1 תכונות הממשק
 - 4.2 עבודה מלא Flow
 - שלב 5: מודלים היברידיים
 - 5.1 Character Model (EfficientNet-based)
 - 5.2 Word Model (CRNN)
 - 5.3 Line Model (TrOCR / Vision Transformer)
 - 5.4 שילוב המודלים
 - שלב 6: אימון מתמשך
 - 6.1 Training Pipeline
 - 6.2 Data Augmentation
 - 6.3 Metrics Dashboard
 - API Endpoints
 - Authentication & Images
 - Image Enhancement

- [Annotation & Training](#)
- [OCR & Stats](#)
- סדר עבודה מומלץ
- תלותיות בין שלבים

name: Hebrew OCR System overview: עם מערכת זיהוי כתב יד בעברית עם Next.js, Express, Firebase. היברידי. תומכת בזיהוי אותיות, מילים ושורות עם אימון שיתופי ML ומודל Firebase. todos:

- id: setup-monorepo content: הקמת מבנה פרויקט - frontend, backend, ml-service status: pending
- id: firebase-setup content: הגדרת Firebase - Firestore, Storage, Auth status: pending
- id: backend-api content: בניית Express API עם routes/controllers/services status: pending
- id: frontend-upload content: Next.js-ממשק העלאת תמונות ב status: pending
- id: image-enhancement content: שיפור וחידוד תמונות עם תצוגה בזמן אמת status: pending
- id: segmentation-service content: שירות Python (שורות/מילים/אותיות) שירות status: pending
- id: annotation-interface content: ממשק תיוג אינטראקטיבי status: pending
- id: character-model content: מודל זיהוי אותיות (EfficientNet) status: pending
- id: word-model content: מודל זיהוי מילים (CRNN) status: pending
- id: line-model content: מודל זיהוי שורות (TrOCR) status: pending
- id: model-fusion content: שילוב תוצאות מכל המודלים status: pending
- id: training-pipeline content: צינור אימון מתמשך status: pending
- id: deployment content: פריסה ופרסום ב GitHub status: pending

מערכת זיהוי כתב יד בעברית

ארכיטקטורה כללית

```
Parse error on line 2:
... subgraph frontend [Frontend - Next.js]
-----^
Expecting 'SEMI', 'NEWLINE', 'SPACE', 'EOF', 'GRAPH', 'DIR',
'TAGEND', 'TAGSTART', 'UP', 'DOWN', 'subgraph', 'end', 'SQE',
'PE', '-)', 'DIAMOND_STOP', 'MINUS', '--', 'ARROW_POINT',
```

```
'ARROW_CIRCLE', 'ARROW_CROSS', 'ARROW_OPEN',  
'DOTTED_ARROW_POINT', 'DOTTED_ARROW_CIRCLE',  
'DOTTED_ARROW_CROSS', 'DOTTED_ARROW_OPEN', '==',  
'THICK_ARROW_POINT', 'THICK_ARROW_CIRCLE', 'THICK_ARROW_CROSS',  
'THICK_ARROW_OPEN', 'PIPE', 'STYLE', 'LINKSTYLE', 'CLASSDEF',  
'CLASS', 'CLICK', 'DEFAULT', 'NUM', 'PCT', 'COMMA', 'ALPHA',  
'COLON', 'BRKT', 'DOT', 'PUNCTUATION', 'UNICODE_TEXT', 'PLUS',  
'EQUALS', 'MULT', got 'SQS'
```

מבנה תיקיות

```
hebrew-ocr/  
├── frontend/                                # Next.js App  
│   ├── src/  
│   │   ├── app/                            # App Router  
│   │   ├── components/                     # React Components  
│   │   │   ├── upload/                     # Upload components  
│   │   │   ├── enhancement/                 # Image enhancement UI  
│   │   │   ├── annotation/                 # Annotation interface  
│   │   │   └── common/                     # Shared components  
│   │   ├── hooks/                          # Custom hooks  
│   │   ├── services/                       # API calls  
│   │   └── types/                          # TypeScript types  
│   └── package.json  
├── backend/                                # Express API  
│   ├── src/  
│   │   ├── routes/                         # API routes  
│   │   ├── controllers/                   # Request handlers  
│   │   ├── services/                      # Business logic  
│   │   │   ├── enhancement.service.ts  
│   │   │   └── websocket.service.ts  
│   │   ├── middleware/                    # Auth, validation  
│   │   └── config/                         # Firebase config  
│   └── package.json  
├── ml-service/                             # Python ML Microservice  
│   ├── models/                             # Trained models  
│   ├── services/  
│   │   ├── enhancement.py                 # Image enhancement pipeline  
│   │   ├── segmentation.py               # Image segmentation  
│   │   ├── character_ocr.py               # Character recognition  
│   │   ├── word_ocr.py                   # Word recognition  
│   │   └── line_ocr.py                    # Line recognition  
│   ├── training/                           # Training scripts  
│   └── requirements.txt  
└── shared/                                # Shared types/utils
```

שלב 1: הקמת תשתית בסיסית

1.1 Backend - Express + Firebase

- הגדרת Firebase Admin SDK
- routes/controllers/services בסיסי עם API מבנה
- Middleware ו rate limiting לאימות
- Upload endpoint לתמונות

1.2 Frontend - Next.js

- הגדרת Firebase Client SDK
- drag & drop ממשק העלאת תמונות עם
- תצוגת תמונות שהועלו

1.3 Firebase Setup

- Firestore collections: `users`, `images`, `annotations`, `training_data`
- Storage buckets: `uploads/`, `processed/`, `models/`
- Security rules

(CRITICAL) שלב 2: שיפור וחיידוד תמונות

שלב קריטי לפני אימון - שיפור איכות התמונות עם תצוגה בזמן אמת.

2.1 Image Enhancement Pipeline

```
Parse error on line 1:
flowchart LR      Ori
^
Expecting 'NEWLINE', 'SPACE', 'GRAPH', got 'ALPHA'
```

פילטרים זמינים 2.2

| פילטר | תיאור | פרמטרים |

|-----|-----|-----|

| Grayscale | המרה לגווני אפור | - |

| Denoise | הסרת רעש | strength: 1-10 |

| Contrast | שיפור ניגודיות | CLAHE clipLimit: 1-5 |

| Binarization | המרה לשחור-לבן | method: Otsu/Adaptive, blockSize, C |

| Deskew | תיקון הטיה | auto-detect angle |

| Sharpen | חידוד | kernel size, strength |

| Morphology | פעולות מורפולוגיות | dilate/erode, kernel size |

2.3 ממשק שיפור תמונות (Real-time)

```
interface EnhancementSettings {
  grayscale: boolean;
  denoise: {
    enabled: boolean;
    strength: number; // 1-10
  };
  contrast: {
    enabled: boolean;
    clipLimit: number; // 1-5
    tileSize: number; // 8-16
  };
  binarization: {
    enabled: boolean;
    method: 'otsu' | 'adaptive' | 'sauvola';
    blockSize: number; // 11-51 (odd)
    constant: number; // 2-20
  };
  deskew: {
    enabled: boolean;
    maxAngle: number; // max rotation degrees
  };
  sharpen: {
    enabled: boolean;
    strength: number; // 0.5-3
  };
}

interface EnhancementPreview {
  originalImage: string; // base64
```

```
enhancedImage: string; // base64
settings: EnhancementSettings;
processingTime: number; // ms
qualityScore: number; // 0-100
}
```

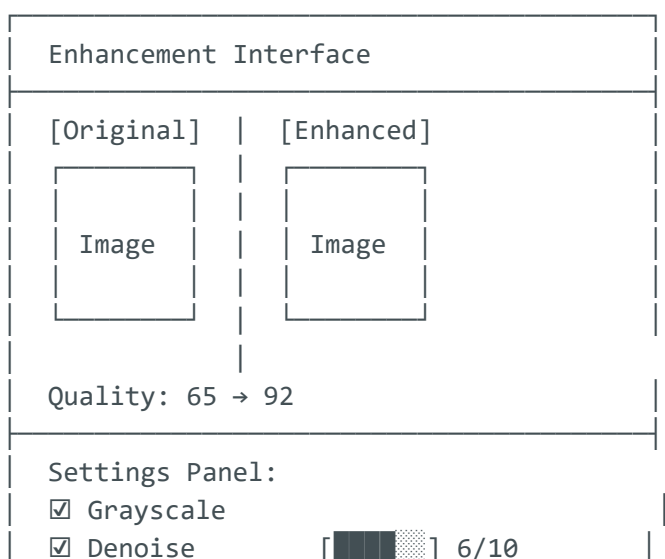
2.4 תכונות ממשק המשתמש

- **תצוגה בזמן אמת:** כל שינוי בפרמטר מעדכן את התצוגה מיידית
- **תמונה מקורית לצד תמונה משופרת Side-by-Side:** השוואה
- **Slider comparison:** גרירה להשוואה על אותה תמונה
- **Zoom & Pan:** התקרבות לפרטים קטנים
- **Undo/Redo:** חזרה לשלבים קודמים
- **Presets:** הגדרות מוכנות מראש לסוגי כתב יד שונים
- **Quality Score:** ציון איכות אוטומטי לתמונה

2.4.1 קומפוננטים נדרשים

```
// frontend/src/components/enhancement/
├─ EnhancementPanel.tsx      // פאנל הגדרות עיקרי
├─ ImageComparison.tsx       // side-by-side השוואה
├─ EnhancementSlider.tsx     // להשוואה Slider
├─ PresetSelector.tsx        // בחירת preset
├─ QualityIndicator.tsx      // אינדיקטור איכות
└─ RealTimePreview.tsx       // תצוגה בזמן אמת
```

מבנה הממשק:



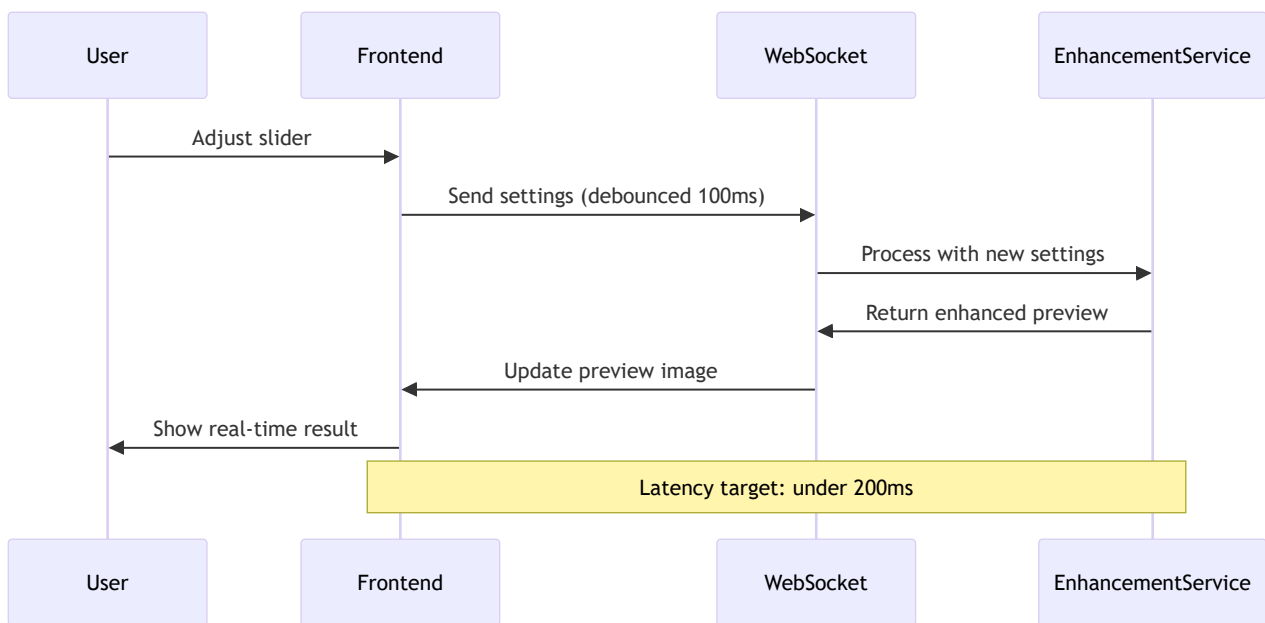
☒ Contrast	[] 3/5
☒ Binarization	[Adaptive ▼]
Block Size:	[] 21
☒ Deskew	[Auto]
☐ Sharpen	[] 2/3

Presets: [דיו בהיר ▼] [Apply]
 [Undo] [Redo] [Reset] [Apply & Save]

מוכנים Presets 2.5

```
const ENHANCEMENT_PRESETS = {
  light_ink: {
    name: 'דיו בהיר',
    settings: { contrast: { clipLimit: 4 }, binarization: { constant: 5 } }
  },
  dark_background: {
    name: 'רקע כהה',
    settings: { binarization: { method: 'adaptive', blockSize: 31 } }
  },
  faded_document: {
    name: 'מסמך דהוי',
    settings: { contrast: { clipLimit: 5 }, sharpen: { strength: 2 } }
  },
  noisy_scan: {
    name: 'סריקה רועשת',
    settings: { denoise: { strength: 8 }, binarization: { method: 'sauvola' } }
  },
  tilted_image: {
    name: 'תמונה מוטה',
    settings: { deskew: { enabled: true, maxAngle: 15 } }
  }
};
```

לתצוגה בזמן אמת WebSocket 2.6



לשיפור תמונות API Endpoints

| Method | Endpoint | תיאור |

|-----|-----|-----|

| POST | [/api/enhance/preview](#) | קבלת תצוגה מקדימה עם הגדרות |

| POST | [/api/enhance/apply](#) | החלת שיפורים ושמירה |

| GET | [/api/enhance/presets](#) | presets קבלת רשימת |

| POST | [/api/enhance/auto](#) | שיפור אוטומטי מומלץ |

| GET | [/api/enhance/quality/:imageId](#) | ציון איכות לתמונה |

שמירת הגדרות לכל תמונה

```

interface ImageEnhancementDoc {
  imageId: string;
  originalPath: string;
  enhancedPath: string;
  settings: EnhancementSettings;
  qualityBefore: number;
  qualityAfter: number;
  appliedAt: Timestamp;
}
  
```


2.9 Python Enhancement Service - מימוש טכני

ספריות נדרשות:

- `opencv-python` - עיבוד תמונה בסיסי
- `scikit-image` - פילטרים מתקדמים
- `numpy` - חישובים
- `Pillow` - מניפולציה על תמונות

פונקציות עיקריות:

```
# ml-service/services/enhancement.py
def apply_enhancement(image: np.ndarray, settings: dict) -> np.ndarray:
    """Apply enhancement pipeline based on settings"""
    if settings['grayscale']:
        image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    if settings['denoise']['enabled']:
        image = denoise_image(image, settings['denoise']['strength'])

    if settings['contrast']['enabled']:
        image = enhance_contrast(image, settings['contrast'])

    if settings['binarization']['enabled']:
        image = binarize_image(image, settings['binarization'])

    if settings['deskew']['enabled']:
        image = deskew_image(image, settings['deskew']['maxAngle'])

    if settings['sharpen']['enabled']:
        image = sharpen_image(image, settings['sharpen']['strength'])

    return image

def calculate_quality_score(image: np.ndarray) -> float:
    """Calculate image quality score (0-100)"""
    # Based on contrast, sharpness, noise level
    pass
```

2.10 אופטימיזציה לביצועים

- **Caching:** שמירת תצוגות מקדימות ב-Redis/Memory
- **Debouncing:** לפני שליחה לשרת ms מעיכוב של 100
- **Image compression:** דחיסת תמונות לפני שליחה (WebP)
- **Batch processing:** עיבוד מספר תמונות במקביל

- GPU acceleration: אם זמין CUDA-שימוש ב

שלב 3: עיבוד תמונות וסגמנטציה

הערה חשובה: שלב זה מתבצע על תמונות משופרות (לאחר שלב 2). שיפור התמונות משפר משמעותית את דיוק הסגמנטציה והזיהוי

3.1 Image Processor

- קבלת תמונה משופרת (לאחר שלב 2)
- חיתוך אוטומטי
- זיהוי שורות (Line Detection)
- זיהוי מילים בתוך שורות (Word Segmentation)
- זיהוי אותיות בתוך מילים (Character Segmentation)

```
Parse error on line 1:
flowchart LR      Enh
^
Expecting 'NEWLINE', 'SPACE', 'GRAPH', got 'ALPHA'
```

3.2 Firestore-מבנה נתונים ב

```
// Image document
interface ImageDoc {
  id: string;
  userId: string;
  originalPath: string;
  enhancedPath?: string;
  status: 'uploaded' | 'enhancing' | 'enhanced' | 'processing' | 'ready' |
'annotated';
  enhancement?: {
    settings: EnhancementSettings;
    qualityBefore: number;
    qualityAfter: number;
    appliedAt: Timestamp;
  };
  segmentation?: {
    lines: LineSegment[];
    words: WordSegment[];
    characters: CharSegment[];
  };
  createdAt: Timestamp;
```

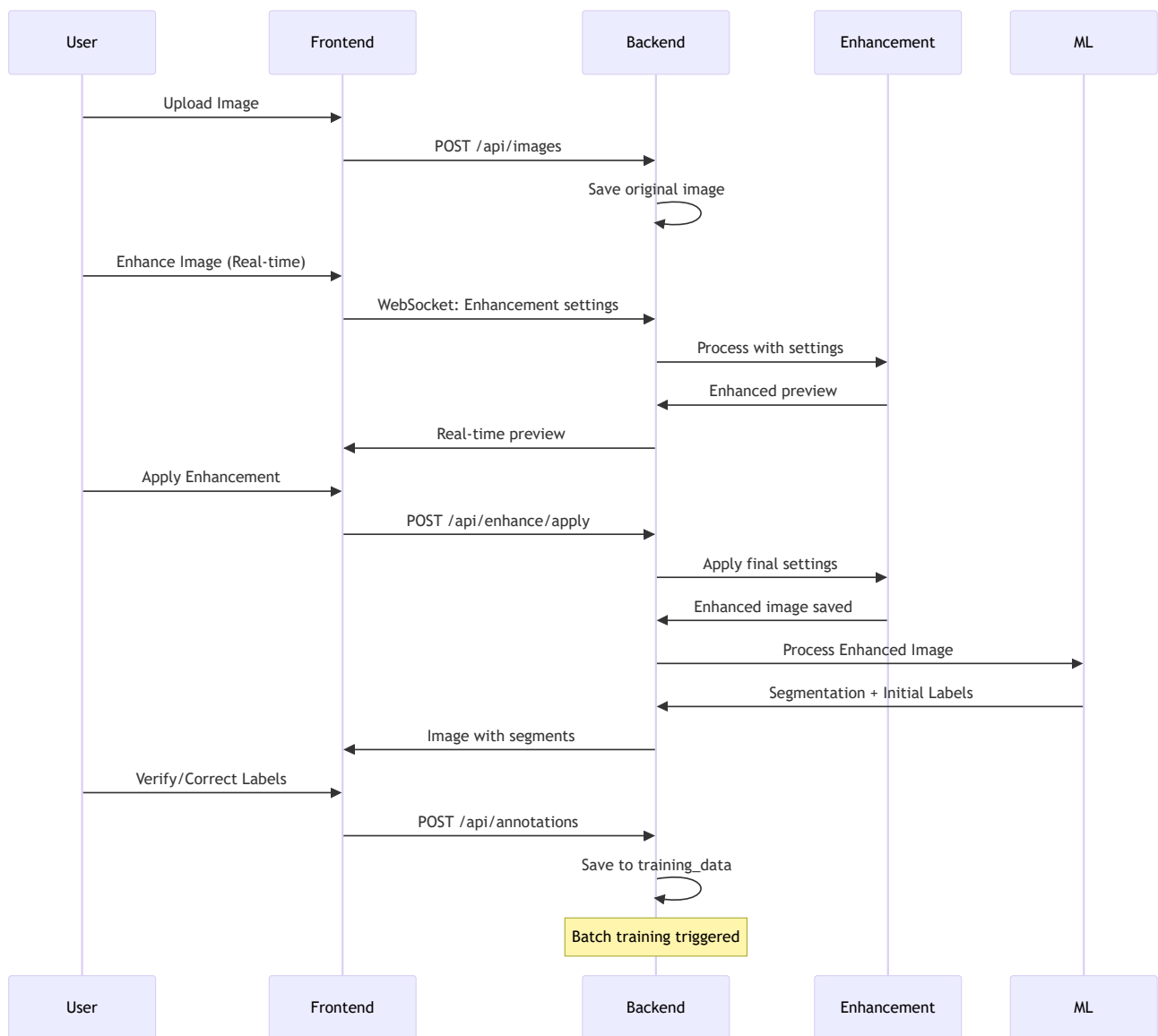
```
}  
  
// Annotation document  
interface AnnotationDoc {  
  id: string;  
  imageId: string;  
  userId: string;  
  type: 'character' | 'word' | 'line';  
  boundingBox: BoundingBox;  
  label: string;  
  confidence: number;  
  isVerified: boolean;  
  createdAt: Timestamp;  
}
```

4: ממשק תיוג (Annotation Interface)

4.1 תכונות הממשק

- bounding boxes תצוגת תמונה עם
- מעבר בין מצבי תיוג: אותיות / מילים / שורות
- מקלדת וירטואלית בעברית
- אישור/תיקון תיוג אוטומטי
- התקדמות ומשוב למשתמש

4.2 עבודה מלא Flow



שלב 5: מודלים היברידיים

5.1 Character Model (EfficientNet-based)

- Fine-tuned EfficientNet-B0
- אותיות + סימני פיסוק 27
- אימון על אותיות מבודדות

5.2 Word Model (CRNN)

- CNN לחילוך features
- RNN (LSTM) לסדר אותיות
- CTC Loss לאימון

5.3 Line Model (TrOCR / Vision Transformer)

- Pre-trained transformer
- על שורות בעברית Fine-tuning
- תמיכה בהקשר מלא

5.4 שילוב המודלים

```
Parse error on line 1:
flowchart TB      Inp
^
Expecting 'NEWLINE', 'SPACE', 'GRAPH', got 'ALPHA'
```

שלב 6: אימון מתמשך

6.1 Training Pipeline

- מאומתות annotations איסוף
- השיפור pipeline שימוש בתמונות משופרות: כל תמונה לאימון עוברת דרך
- training batches יצירת
- incremental אימון למודלים
- שמירת model versions

זרימת נתונים לאימון:

```
Parse error on line 1:
flowchart LR      Ann
^
Expecting 'NEWLINE', 'SPACE', 'GRAPH', got 'ALPHA'
```

6.2 Data Augmentation

לאימון augmentation בנוסף לשיפור התמונות, יש להוסיף:

- Rotation (± 5 degrees)

- Scaling (0.9-1.1)
- Brightness adjustment
- Gaussian noise
- Elastic distortion

6.3 Metrics Dashboard

- דיוק לפי סוג (אות/מילה/שורה)
- כמות נתונים מתויגים
- התקדמות אימון
- **Impact of enhancement:** השוואת דיוק עם/בלי שיפור

API Endpoints

Authentication & Images

| Method | Endpoint | תיאור |

|-----|-----|-----|

| POST | </api/auth/register> | הרשמת משתמש |

| POST | </api/images/upload> | העלאת תמונה |

| GET | </api/images/:id> | קבלת תמונה עם segmentation |

Image Enhancement

| Method | Endpoint | תיאור |

|-----|-----|-----|

| POST | </api/enhance/preview> | קבלת תצוגה מקדימה עם הגדרות |

| POST | </api/enhance/apply> | החלת שיפורים ושמירה |

| GET | </api/enhance/presets> | קבלת רשימת presets |

| POST | </api/enhance/auto> | שיפור אוטומטי מומלץ |

| GET | /api/enhance/quality/:imageId | ציון איכות לתמונה |

| WebSocket | /ws/enhance | תצוגה בזמן אמת |

Annotation & Training

| Method | Endpoint | תיאור |

|-----|-----|-----|

| POST | /api/annotations | שמירת תיוג |

| GET | /api/annotations/:imageId | קבלת תיוגים לתמונה |

OCR & Stats

| Method | Endpoint | תיאור |

|-----|-----|-----|

| POST | /api/ocr/recognize | זיהוי טקסט בתמונה |

| GET | /api/stats | סטטיסטיקות מערכת |

סדר עבודה מומלץ

- שבוע 1: הקמת תשתית בסיסית (Firebase, Express, Next.js)
- שבוע 2: Upload flow + Image storage
- שבוע 3: Image Enhancement Service (Python) + Real-time UI
- שבוע 4: WebSocket integration + Enhancement pipeline
- שבוע 5: Segmentation service (Python)
- שבוע 6: Annotation interface
- שבוע 7: Character model + training
- שבוע 8: Word/Line models
- שבוע 9: Model fusion + API
- שבוע 10: Testing, optimization, deployment

תלותיות בין שלבים

