In manifest:

<uses-feature android:name="android.hardware.Camera"  
 android:required="true"></uses-feature>

<uses-permission  
 android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"  
 android:maxSdkVersion="18" />

In main:

CODE = 101;

public void takePhoto(View view) {  
 Intent intent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);  
 if (intent.resolveActivity(getPackageManager())!= null){  
 startActivityForResult(intent, CODE);  
 }  
}  
  
@Override  
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) { - handle result  
 super.onActivityResult(requestCode, resultCode, data);  
 if (requestCode == CODE && resultCode == RESULT\_OK){  
 Bundle bundle = data.getExtras();  
 Bitmap bitmap = (Bitmap)bundle.get("data");  
 imageView.setImageBitmap(bitmap);  
 }  
  
}

**Saving photos in high quality**

In manifest in activity add:

<provider  
 android:name="android.support.v4.content.FileProvider"  
 android:authorities="com.example.android.fileprovider"  
 android:exported="false"  
 android:grantUriPermissions="true">  
 <meta-data  
 android:name="android.support.FILE\_PROVIDER\_PATHS"  
 android:resource="@xml/file\_path" /> - new created xml resource file   
</provider>

Create new resource file xml type with paths root and set:

<external-path  
 name="my\_images" - name  
 path="Android/data/com.example.boris.myapplication/files/Pictures"/> - where it will download (default path, begins with package)

In main:

private File getImageFile() throws IOException { - function for getting location  
 String date = new SimpleDateFormat("yyyyMMdd").format(new Date());  
 String name = "jpg\_" + date +"\_";  
 File dir = getExternalFilesDir(Environment.DIRECTORY\_PICTURES); - get default dir  
  
 File imageFile = File.createTempFile(name, ".jpg",dir);  
 curreinImagePath = imageFile.getPath(); - not connected, but you can create global variable with path and in each photo set it here   
 return imageFile;  
}

public void takePhoto(View view) {  
 Intent intent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);  
 if (intent.resolveActivity(getPackageManager())!= null){ - avoid exceptions   
 File imageFile = null;  
  
 try {  
 imageFile = getImageFile();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 if (imageFile != null){  
 Uri imageUri = FileProvider.getUriForFile(this, "com.example.android.fileprovider", imageFile);  
 intent.putExtra(MediaStore.EXTRA\_OUTPUT, imageUri);  
 startActivityForResult(intent, CODE);  
 }  
 }  
}

Video

In manifest:

<uses-feature  
 android:name="android.hardware.camera2"  
 android:required="true" />

In main:

Int VIDEO\_CODE = …; - for checking

Intent intent = new Intent(MediaStore.ACTION\_VIDEO\_CAPTURE);  
if (intent.resolveActivity(getPackageManager()) != null){  
 startActivityForResult(intent,VIDEO\_CODE);  
}

@Override  
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

super.onActivityResult(requestCode, resultCode, data);  
 if (requestCode == VIDEO\_CODE && resultCode == RESULT\_OK){ -   
 Uri videoUri = data.getData();  
 Intent displayVidos = new Intent(this, VideoActivity.class);  
 displayVidos.putExtra("dir", videoUri.toString());  
 startActivity(displayVidos);  
 }  
}

To make phone reading your text

implements TextToSpeech.OnInitListener

int RECORD\_CODE = 3;  
TextToSpeech textToSpeech = null;  
EditText textView = …;

Intent intent = new Intent(TextToSpeech.Engine.ACTION\_CHECK\_TTS\_DATA);  
startActivityForResult(intent, RECORD\_CODE); - handled event for example button

@Override  
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) { - handle event  
 super.onActivityResult(requestCode, resultCode, data);  
 if (requestCode == RECORD\_CODE && resultCode == TextToSpeech.Engine.CHECK\_VOICE\_DATA\_PASS){  
 textToSpeech = new TextToSpeech(this, this); - set implemented method  
 }  
}

@Override  
public void onInit(int status) {  
 if (status == TextToSpeech.SUCCESS){

int language\_status = textToSpeech.setLanguage(Locale.US);  
 if (language\_status == TextToSpeech.LANG\_MISSING\_DATA || language\_status == textToSpeech.LANG\_NOT\_SUPPORTED){  
 Toast.makeText(getApplicationContext(), "incorrect language", Toast.LENGTH\_SHORT).show();  
 }else {  
 String data = textView.getText().toString();  
 int speachStatus = textToSpeech.speak(data, TextToSpeech.QUEUE\_FLUSH, null); - start reading   
 if (speachStatus == TextToSpeech.ERROR){  
 Toast.makeText(getApplicationContext(), "error while recognizing speach", Toast.LENGTH\_SHORT).show();  
 }  
 }  
  
 }else {  
 Toast.makeText(getApplicationContext(), "recording failed", Toast.LENGTH\_SHORT).show();  
 }  
}